MICROCOMPUTADOR CURSO PRÁTICO

TIRE O MÁXIMO DO MICRO EM 30 SEMANAS

PUBLICAÇÃO SEMANAL ILUSTRADA Cr$ 6.500

3

Manutenção do micro: faça você mesmo

Quebra de sigilo: o acesso pirata

MANAUS, ALTAMIRA, MACAPÁ, JI.PARANÁ, PORTO VELHO, RIO BRANCO, SANTARÉM, BOA VISTA, SINOP, ALTA FLORESTA: Cr$ 8.450

rioGráfica

MICROCOMPUTADOR CURSO PRÁTICO

APLICAÇÕES

Descobrindo a senha 57

PERIFÉRICOS

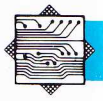
Cópias quentes 59

SOFTWARE

Arquivo organizado 62

PROJETOS DE PROGRAMAS

Gráficos 64

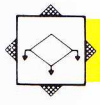


RAIO X

Micros Sharp 66

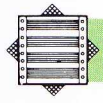
OFICINA

Ferramentas essenciais 68



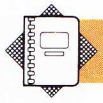
LINGUAGENS

Duplo desempenho 70



TÉCNICAS DE PROGRAMAÇÃO

Conceitos básicos 75



PERFIL

Terminal bancários 76

Nota: Muitos dos equipamentos que aparecem na seção RAIO X não são normalmente comercializados no Brasil. Sua divulgação é feita a título de informação ao leitor.

TESTE

Depois de ler com atenção todo o fascículo, responda a estas perguntas. Veja a solução no próximo número, junto com um novo teste.

1) Qual o tipo de impressora que necessita de papel especial metalizado?
2) Qual a vantagem de usar algoritmos genéricos, em vez de específicos ao programa?
3) Qual o único tipo de acesso usado nos arquivos gravados em fita?
4) Onde se armazenam os dados no computador de bolso Sharp PC-1251?
5) Qual o comando LOGO que impede que a tartaruga saia fora dos limites da tela?

RESPOSTAS DO TESTE ANTERIOR

1) O Apple II *c* não possui nenhum slot, pois as conexões para periféricos estão todas embutidas em seu hardware básico.
2) Segundo o algoritmo proposto, um quadrado mágico deve necessariamente ter um número ímpar de quadrados de lado.
3) O BASIC só distingue as duas primeiras letras de cada variável; portanto, CONT e CONTADOR seriam por ele identificadas como a mesma variável.
4) Em relação ao sistema de armazenamento e acesso, os arquivos podem ser binários, seqüenciais (seriais) ou aleatórios.
5) A medida da resolução gráfica de um monitor de vídeo é o pixel. Quanto mais pixels, mais alta será a resolução.
6) Num procedimento LOGO para desenhar um triângulo eqüilátero, a tartaruga deve virar 120°, e não 60°. Portanto, o procedimento correto é:
   REPITA 3 [FR 50 DI 120]

PLANO DA OBRA

MICROCOMPUTADOR - CURSO PRÁTICO é uma coleção de 31 fascículos de periodicidade semanal. Os trinta primeiros terão, cada um, vinte páginas internas (miolo) e quatro capas, além de uma introdução de dezesseis páginas acompanhando o fascículo 1. O fascículo 31 terá doze páginas e quatro capas, e incluirá o índice geral da obra. Os miolos, encadernados, constituem dois volumes de um curso prático de microcomputação, com seções indicadas por tarjas de diferentes cores.
As duas primeiras capas são descartáveis; as 3.ªs e 4.ªs trazem programas de jogos. Encadernadas, formam um volume de cem páginas.

COMO ENCADERNAR

As capas duras, incluindo as guardas, estarão à venda simultaneamente com os fascículos 10 (**Volume 1**) e 25 (**Volume 2** e **Jogos**).
Volume 1 - Deve ser encadernado com os elementos dispostos nesta ordem: guardas; frontispício e introdução (publicados com o fascículo 1); miolos dos fascículos de 1 a 15; e guardas.
Volume 2 - Guardas; frontispício (que virá com o fascículo 31); miolos dos fascículos de 16 a 31; e guardas.
Jogos - Guardas; frontispício e sumário (que constituem as 3.ªs e 4.ªs capas do fascículo 31); 3.ªs e 4.ªs capas dos fascículos de 1 a 30; e guardas.

rioGráfica **Diretoria Executiva:** Oscar Neves; Filipe Zander; Nilo Sérgio de Almeida; Danilo Esteves Costa; **GRUPO EDUCAÇÃO E CULTURA - Planejamento e Comercialização - Diretor:** Roberto Combochi; **Gerente de Marketing:** Jaime Rodrigues; **Gerente de Produto:** José Mauro Porto; **Gerente do P.C.P.:** Aylton Menezes; **Gerente de Vendas:** Rubens Barbosa; **Coordenadores de Promoção:** Paulo Cesar M. Seixas; Edgar Mello M. Neto; **Gerente de Circulação:** Norberto Martin; **Supervisor de Operações:** Abel Pereira Pinto - **Editorial - Diretor:** João J. Noro; **Editor-chefe:** Maurício Rittner; **Administração Editorial:** Heitor de Souza Paixão; **Editor:** Jesse Navarro; **Revisores:** Isis Augusta Loyolla; Cacilda Guerra; Luiz Vicente Vieira Filho; Marcos Domingos Agathão; Maria da Graça Mendonça Couto; **Diretor de Arte:** Aníbal Monteiro; **Chefe de Arte:** Bonifácio Duardes Miranda; **Chefe de Estúdio:** José Yuji Kuribayashi; **Assistente de Arte:** Miguel Luis Escámez Simón - **Colaboradores - Editores-assistentes:** Isa Mara Lando; Luiz Carlos Pizarro Marin; Cláudio Marcondes; **Texto:** Jussara Maturo; **Preparação de Texto:** Luiz Carlos Cardoso; **Tradução:** Cláudio Marcondes; Lucy Morais R. Petroucic; Maria Clara Cescato; Patrícia de Paiva e Castro; Regina Carmo R. Silva; **Pesquisa:** Stella Maria Quentel; **Arte:** Toshio Morobosi; **Assessoria Técnica:** Derval Junqueira de Aquino; Lauro de Lauro; Luiz Henrique Alayon. As fotos não creditadas pertencem à obra original. Segue-se o crédito das demais: Fernando Scavone; Itautec. © APSIF Copenhagen, 1984; © Orbis Publishing Ltd., 1984; © Editora Rio Gráfica Ltda., 1985, para a língua portuguesa. **Composição:** ERG Ltda. e AM Produções Gráficas Ltda. **Impressão:**

NÚMEROS ATRASADOS - Você poderá comprar os exemplares que faltam em sua coleção, pelo preço do último fascículo posto à venda, em sua banca de jornal preferida. Caso não consiga obtê-los, faça sua solicitação por carta endereçada diretamente à Editora Rio Gráfica Ltda., rua Itapiru, 1209, Rio Comprido, Rio de Janeiro, CEP 20251. O atendimento será feito por via postal. Nas cidades do Rio de Janeiro e de São Paulo, as compras poderão ser efetuadas pessoalmente nos seguintes endereços: Rio de Janeiro - Rua Itapiru, 1209, Rio Comprido; São Paulo - Rua Frei Caneca, 1152, Consolação.

Distribuidor exclusivo para todo o Brasil: Fernando Chinaglia - Rua Teodoro da Silva, 907, Rio de Janeiro. Distribuidor para Portugal: Electroliber Lda. - Rua Prof. Reinaldo dos Santos, 1488, Lisboa.

DESCOBRINDO A SENHA

A maior parte dos casos de acesso ilegal aos computadores de grande porte por meio de microcomputadores e modems envolve órgãos governamentais e corporações multinacionais.

O filme *Jogos de guerra (War games)* fascinou a imaginação de muitos usuários de microcomputador. O personagem principal, usando um micro e um modem, consegue acesso ilegal a vários computadores, altera resultados dos exames de sua faculdade, reserva bilhetes aéreos e copia os jogos mais atuais. Mas as conseqüências da brincadeira tornam-se dramáticas quando ele acessa o computador do NORAD, órgão responsável pela defesa aérea dos Estados Unidos, e quase inicia uma guerra nuclear.

Distribuição ilegal

Embora negue o incidente, a Pepsi-Cola foi vítima de pirataria: alguém dos Estados Unidos obteve acesso ilegal ao computador dessa companhia americana no Canadá. Os piratas enviaram grandes carregamentos da bebida a várias localidades, transferindo quantias muito altas de dinheiro para suas contas.

Boa parte das violações de computador tem como autores adolescentes que usam micros domésticos e modems. Qualquer computador que permita acesso por telefone torna-se vulnerável. Em 1983, a realidade chegou perto da ficção, quando se suspeitou que dois "piratas" haviam acessado a rede de computadores do NORAD em Omaha, Nebrasca, nos EUA.

Os envolvidos eram dois adolescentes de Los Angeles, que violaram a Arpanet — rede secreta de computadores do Departamento de Defesa dos EUA. Valendo-se de um VIC-20 da Commodore e de um TRS-80 da Tandy, eles conseguiram dados de vários computadores da Arpanet. Esses equipamentos, em geral, pertencem a empresas contratadas pelo Departamento de Defesa, órgãos de pesquisa e universidades. Como o sistema compartilha dados científicos, nenhuma informação confidencial foi obtida. A facilidade com que os adolescentes violaram a Arpanet, porém, provocou um enorme constrangimento no Departamento de Defesa.

A facilidade de acesso, em muitos casos, não se deve a falhas do computador, mas à preguiça humana. Os usuários da Arpanet talvez não tivessem senhas muito criativas. Os meninos partiram da suposição correta de que a Universidade de Berkeley fosse usuária da Arpanet e entraram na rede com a óbvia senha UCB (Universidade da Califórnia em Berkeley). Com isso, ficaram livres para acessar qualquer computador da Arpanet, inclusive a instalação subterrânea central do NORAD, em Omaha.

As conseqüências da "invasão" apenas não foram mais sérias porque o escritório central do NORAD está ligado à Arpanet. Todos os computadores responsáveis pela defesa aérea do país foram instalados sob as montanhas Cheyne, no Colorado, e não se conectam às linhas públicas de telefone.

Mas há vários computadores que, ao contrário do sistema do NORAD, estão sujeitos a violações de piratas. Em julho de 1983, um grupo de adolescentes de Milwaukee acessou mais de sessenta computadores de universidades e corporações, além do Laboratório Nacional de Los Alamos, destinado à produção de armas. Mais uma vez, segundo as autoridades, nenhuma informação confidencial foi obtida. Os meninos vasculharam apenas registros, relatórios de rotina e mensagens. Acionou-se o FBI (Federal Bureau of Investigations) para descobrir como o grupo, autodenominado "414", realizou a façanha. O grupo defendeu-se afirmando que não havia medidas de segurança em nenhum dos computadores para os quais eles tinham telefonado.

Muitos outros casos não se tornam públicos porque as organizações evitam admitir que seus sofisticados computadores foram invadidos por garotos com micros. Muitas vezes, as próprias organizações desconhecem uma infiltração, dada a dificuldade de se detectar um usuário não autorizado — embora piratas mais insolentes deixem mensagens do tipo "vocês não me pegam" e se assinem "Destruidor de Sistemas" ou "Capitão Gancho".

Como ocorre a pirataria? Um pirata em potencial necessita apenas de um micro, um modem e engenhosidade. O primeiro obstáculo consiste em descobrir o número de telefone do computador. No caso de redes de acesso público — como a Telecom Gold, na Inglaterra, ou The Source, nos EUA —, isso não é difícil, já que os números são de domínio público. A pirataria se complica quando se trata de compu-

tadores particulares. Porém, dispondo-se pelo menos de uma vaga idéia da localização do computador, a técnica que o protagonista de *Jogos de guerra* empregou pode ser útil. Ele programou seu micro para discar todos os números possíveis em sua cidade. Se algum computador respondesse com um som característico, o micro registrava o número. Mas, se uma pessoa atendesse à chamada, o modem desligava e passava para o número seguinte. Para obter resultados mais rápidos, o jovem "herói" do filme usou um modem de discagem automática.

Uma vez realizada a conexão, haverá uma senha a responder. Certas redes permitem acesso limitado quando se digita "VISITANTE" ou "NOVO USUÁRIO" ou apenas se pressiona a tecla [RETURN]. No entanto, o verdadeiro pirata procura descobrir a senha por tentativa e erro. O que não é difícil, devido à falta de imaginação dos usuários, que utilizam como senha seu próprio nome ou palavras óbvias — "SEGREDO" e "SENHA", por exemplo. Quando recorrem a algarismos, a preferência recai em seqüências fáceis de lembrar, como a data de nascimento. Muitos computadores permitem várias tentativas antes de desconectar o intruso. E, mesmo assim, sem levantar suspeitas, pode-se ligar de novo para o computador.

Uma vez dentro do sistema, a maior parte dos piratas apenas examina os registros, vasculha os programas de jogos e "conversa" com outros piratas que estão na linha. Mas há os destruidores, que apagam registros, deixam mensagens obscenas e tentam bloquear todo o sistema. Uma invasão de sistema pode trazer conseqüências desastrosas para os usuários legítimos.

Enfrentar a máquina

Mesmo nos computadores de grande porte mais sofisticados, os programadores sempre deixam entradas de emergência no sistema para — sem passar pelas medidas de segurança — penetrar com rapidez no programa. Muitas vezes, esse recurso é desconhecido até pelos operadores do sistema.

A maior parte dos incidentes ocorre com computadores de universidades, pois eles permitem discagem externa e operam com acesso aberto. Contando com milhares de usuários, o acesso aberto é o meio mais prático para a operação desses sistemas. Só que, assim, ficam muito mais vulneráveis à pirataria.

Após entrar no sistema, os intrusos podem pular de um computador a outro, passando por usuários legítimos. Um estudante do campus da Universidade da Califórnia em San Jose encontrou uma brecha no projeto Talk, que permite aos estudantes conversarem, via computador e modem, com outros campi da Universidade da Califórnia. Vencendo o obstáculo local, o rapaz conseguiu comunicar-se com computadores de todo o país e mesmo do exterior.

Os motivos da pirataria, em geral, se resumem à emoção de vencer o sistema. Muitos piratas têm seu código de ética particular e não danificam o sistema, divertindo-se apenas com a descoberta dos códigos. Esquecem, no entanto, que usar o tempo do computador sem pagar também é ilegítimo.

Os bancos sempre foram muito atingidos pelos crimes por computador, envolvendo a transferência de dinheiro para contas falsas. A dificuldade do cálculo exato dos prejuízos causados pela pirataria deve-se ao desinteresse das companhias em admitir publicamente que foram vítimas de fraudes por computador. O aumento no número de usuários de microcomputador e de redes de computadores que usam linhas telefônicas agravou ainda mais os problemas desse gênero. Os métodos de fraude são os mesmos, não importando tratar-se de adolescentes travessos, apagando registros e roubando tempo do computador, ou criminosos desviando dinheiro para as próprias contas.

Tentativa e erro
O método usado pelos piratas envolve várias tentativas, até que, em raras ocasiões, se obtém entrada não autorizada no sistema. Ainda que consiga localizar o computador, para validar sua entrada no sistema, o pirata depara com o seguinte diálogo:

```
CONNECT 0X001001 14:32
     12/7/84
>
USER?
>HELP
USER?
>ESOFF
USER?
>UK001
PASSWORD?
>GUEST
PASSWORD?
>NEWUSER
PASSWORD?
>QWERTY
LOGOFF 15.13 CONNECT
     TIME = 0.13 MINS

CONNECT BYF990
     15.14.02 12/07/84
>
USER?
>UK001
PASSWORD?
>SYSOP
LOGON 15.15.07 12/07/84
HOST: BYF990/SRYLOM
USER: UK001
SERIAL NO: ZA100-7
PRIORITY: SUPERUSER
STATUS: ACTIVE
YOU ARE SYSOP
7 USER(S)
APP01 APP02 BYF7 BTY04
     BZX88 BZX02 SYSOP
>REMOVE ARP01
USERS(S) ARP01
     DISCONNECTED
>WHO IS BTY04
BTY04     CAREY DIMMIT,
     BTY LOPP,  742
     SILICON DV
HERTS 07662-093164
```

Pressionando [RETURN]: usuário, identifique-se
Nenhuma ajuda: computador não amigável
Primeira tentativa de identificação válida
Entrada invalidada
Segunda tentativa de identificação
Identificação aceita: pede a senha
Primeira tentativa
Recusa: segundo pedido
Segunda tentativa
Recusa: terceiro pedido
Tentativa desesperada

Depois da terceira tentativa fracassada de achar a senha, desconecta-se o usuário.

Pressione [RETURN]
Identificação válida: pede a senha
Primeira tentativa operador de sistema
Número de série do programa
Arquivos liberados
Usuário desconectado se não usar o sistema de maneira correta
Confirmação
Liste todos os usuários
Comando reservado ao operador de sistema
Usuário válido desconectado

Acerto ocasional
Algumas vezes, depois de várias tentativas ou por pura sorte, o pirata descobre a senha válida e entra no sistema. O diálogo fictício acima descreve como o usuário poderia obter informações confidenciais sobre um usuário legítimo.

CÓPIAS QUENTES

Para as atividades comerciais, as impressoras matriciais ou tipo margarida são as mais indicadas, por sua boa qualidade de impressão. Os usuários particulares contam com alternativas mais baratas.

Todo usuário precisou, em algum momento, de cópias impressas de texto ou de listagem de programas — e constatou o alto custo de uma boa impressão. As impressoras matriciais chegam a ser duas ou três vezes mais caras do que o próprio computador. E a compra de uma impressora tipo margarida não se justifica para a maioria dos usuários.

As impressoras térmicas, porém, representam uma alternativa de baixo custo. Para impressão, tais equipamentos dispensam o impacto de agulhas ou letras metálicas sobre a fita. A ação do calor grava os dados no papel. Para tanto, é indispensável o uso de papel termossensível.

Esse tipo de impressão requer uma matriz de elementos que, quando requisitados, se aquecem, fazendo a pequena área correspondente do papel mudar de cor.

A base do processo das impressoras matriciais está no impacto provocado pela matriz de agulhas, suportada por uma cabeça, que se move horizontalmente pela página, imprimindo os dados de forma barulhenta. Nesse sentido, as impressoras térmicas são mais vantajosas, dada a sua operação silenciosa. A Apple adotou a nova tecnologia e desenvolveu uma impressora térmica para seu computador, batizando-a de Silentype ("tipo silencioso").

As impressoras eletrostáticas representam um tipo intermediário entre as matriciais e as térmicas: não são tão barulhentas como as primeiras nem tão silenciosas quanto as térmicas. Tiveram larga aceitação no exterior, quando as indústrias Sinclair utilizaram o sistema em sua impressora ZX, vendida em grandes quantidades como periférico dos micros ZX-81 e Spectrum-ZX.

A cabeça de impressão das eletrostáticas compõe-se de um fio que se move sobre o papel recoberto de uma camada metálica. A cada caractere acionado, o circuito da impressora gera uma faísca que queima a camada metálica, revelando o papel preto do fundo. A Sinclair aperfeiçoou o sistema ao empregar duas cabeças ou dois fios para impressão. Mas ainda assim precisa de oito movimentos da cabeça para formar cada linha de caracteres. Como a impressora ZX imprime apenas 32 caracteres por linha, a velocidade de impressão é aceitável.

Impressora térmica Tandy
Com mais facilidade de adaptação do que as outras impressoras de baixo custo, a TP-10 da Tandy é compatível com toda a linha de microcomputadores dessa empresa.

A maior desvantagem das impressoras térmicas e eletrostáticas é o seu papel especial. Além de caros, tais papéis só estão disponíveis em rolos, o que dificulta o armazenamento.

A impressora térmica exige cuidados especiais na compra do papel, que deve ser adquirido no grau correto de sensibilidade. Caso contrário, a imagem não será revelada a contento, desbotando com o tempo ou com a exposição ao calor.

O papel eletrostático é ainda mais delicado e, se mãos úmidas o pegarem, dissolve-se a cobertura, borrando ou apagando a imagem. Nos dois casos, o meio mais indicado de garantir a durabilidade da imagem é tirar fotocópias.

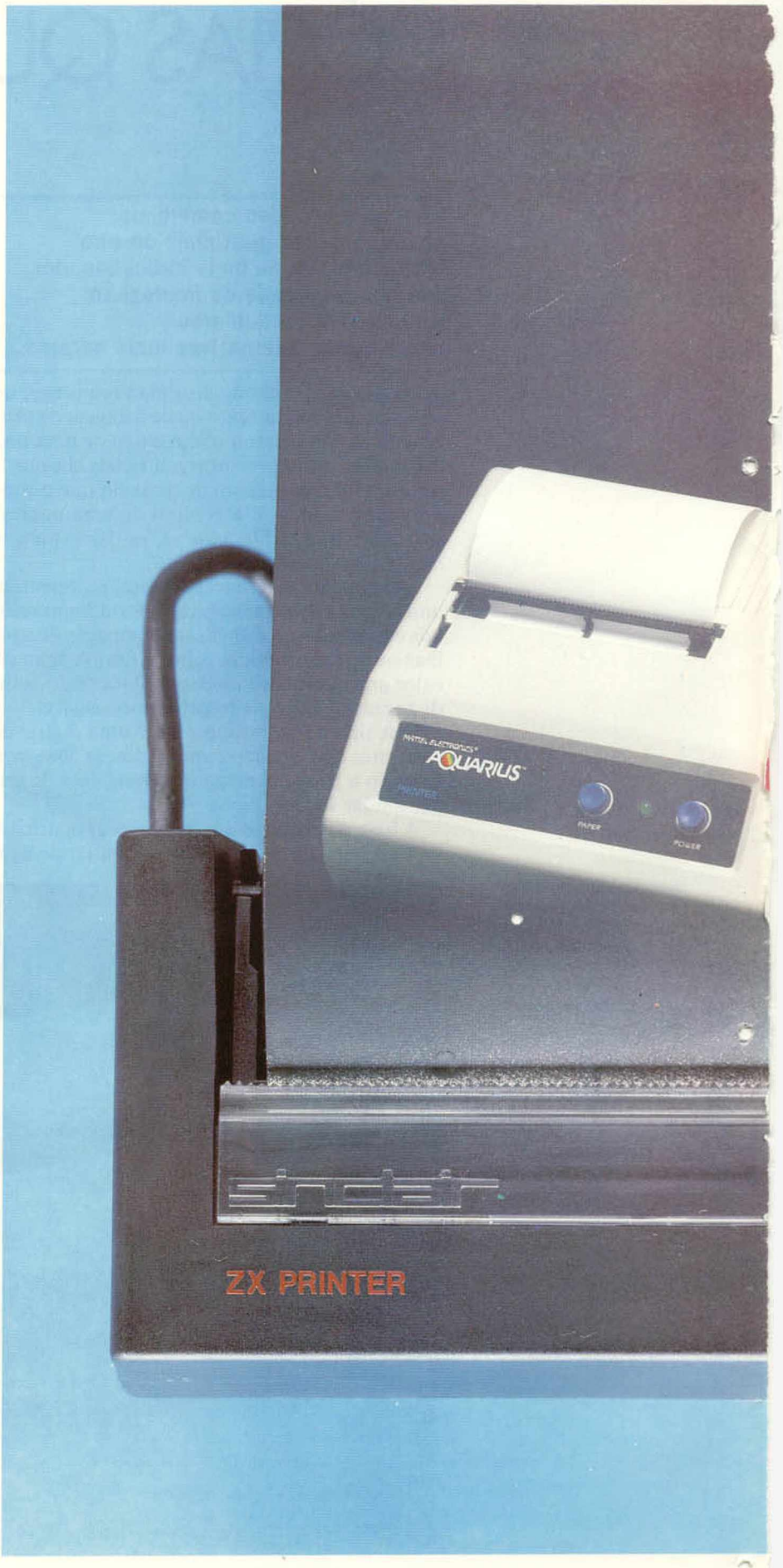
Apesar dos inconvenientes, esses dois tipos de impressora são a melhor alternativa para os usuários de microcomputador. Elas permitem que as cópias sejam feitas diretamente da tela, produzindo textos e gráficos, embora a qualidade de impressão não seja perfeita.

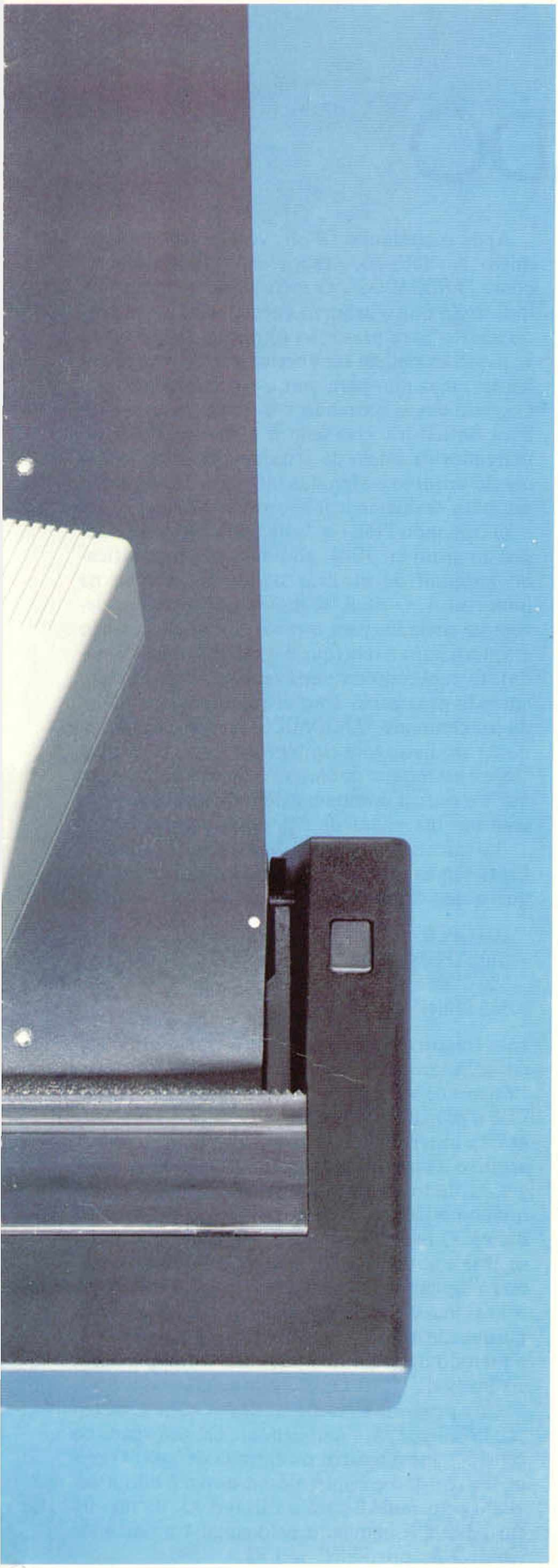
Por outro lado, os plotters têm-se revelado a mais nova alternativa para sistemas de impressão. Com uma ou mais canetas, eles desenham os caracteres como se usassem canetas esferográficas. Os plotters também são indicados na impressão de gráficos.

Sua grande vantagem está na utilização de papel comum. Além disso, seu custo não supera o das impressoras térmicas e eletrostáticas.

Apesar destas novas técnicas de impressão, os sistemas sem impacto continuam, com seu mecanismo simples e limitado, a ser a opção mais barata para o usuário de microcomputadores.

Big Brother
Além de mais barata, a impressora térmica Big Brother EP-22, de fabricação japonesa, incorpora três funções: trabalha como impressora térmica e como máquina de escrever portátil projetada para operar com papel térmico, além de permitir a impressão normal de impacto com fita. A EP-22 armazena em sua memória, mantida com bateria, até 2.000 caracteres, conservando os textos gravados mesmo quando desligada. O conteúdo da memória, no entanto, não pode ser totalmente modificado. Só é permitida a correção dos últimos dezesseis caracteres inseridos, pois o texto, antes de ser impresso, aparece para o usuário num visor de cristal líquido (LCD). A qualidade da impressão também não se compara à de uma impressora tipo margarida. As letras minúsculas, por exemplo, são cortadas nas extensões inferiores. Por outro lado, sendo uma impressora portátil para acompanhar microcomputadores, a EP-22 executa tarefas difíceis de realizar em qualquer tipo de impressora de impacto.

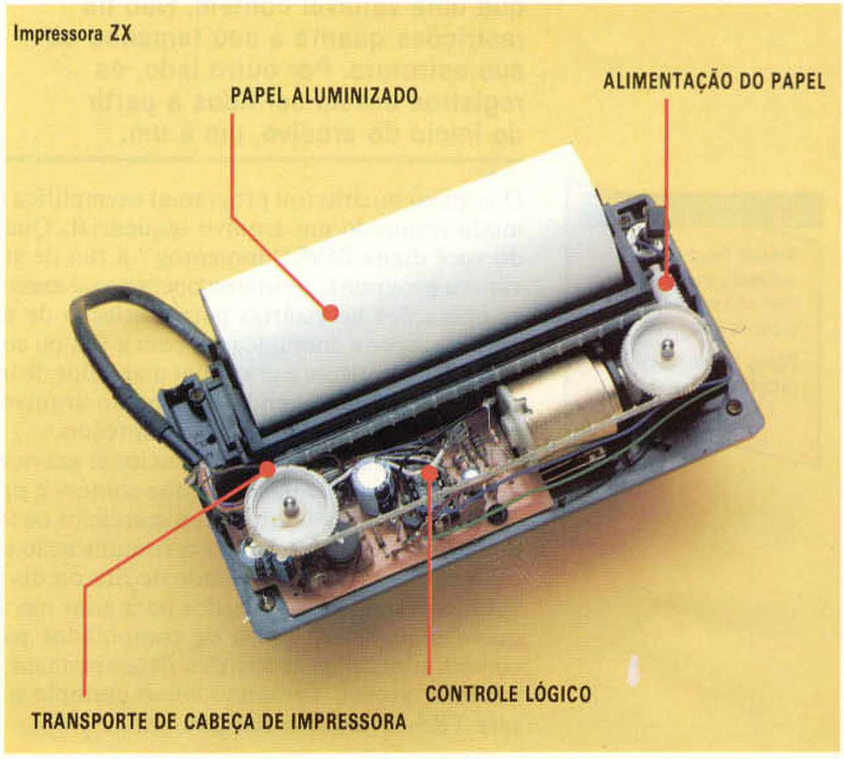




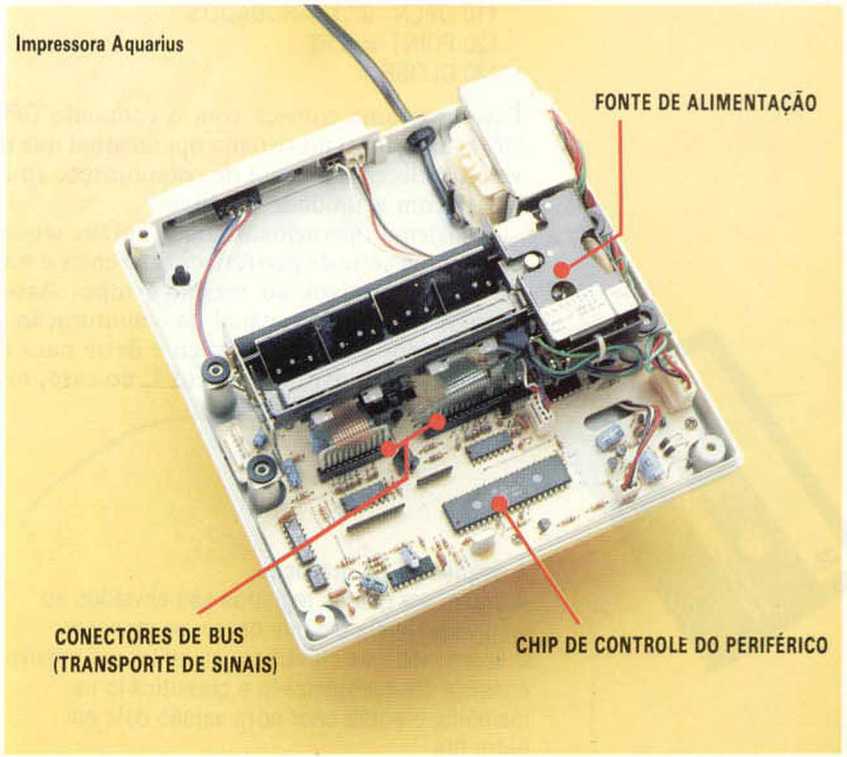
Impressora ZX
Introduzida como periférico do micro Sinclair ZX-81, a impressora ZX também é compatível com o Spectrum-ZX. Comercializada como uma alternativa econômica para a impressão de listagens de programas e textos, a ZX apresenta a desvantagem de funcionar apenas com papel eletrostático.

Impressora Aquarius
Apesar de dispor de econômica impressora de quatro cores, que usa canetas esferográficas, a Aquarius adotou a impressora térmica como opção de baixo custo para os usuários de seus microcomputadores. Mas, assim como a ZX, a Aquarius depende de papel especial.

Impressora ZX



Impressora Aquarius



ARQUIVO ORGANIZADO

Arquivos seqüenciais armazenam tudo o que uma variável contém. Não há restrições quanto a seu tamanho ou sua estrutura. Por outro lado, os registros devem ser lidos a partir do início do arquivo, um a um.

OPÇÃO E SEQÜÊNCIA

Default: Opção pré-assumida, que prevalece caso não se faça outra especificação.

String: Seqüência de caracteres alfanuméricos.

O arquivo binário (ou programa) exemplifica de modo resumido um arquivo seqüencial. Quando você digita SAVE "nomeprog" a fim de gravar seu programa, o sistema operacional executa as operações necessárias para a criação de um arquivo: abre a comunicação com a fita ou com a unidade de discos e grava um marcador de início de arquivo, contendo o nome do arquivo e algumas informações sobre o conteúdo.

Em seguida, o sistema operacional grava no arquivo o bloco de memória que contém o programa atual; depois, grava um marcador de fim de arquivo; e afinal encerra a comunicação entre o computador e a unidade de fita ou disco.

Os comandos BASIC usados para criar um arquivo seqüencial variam de computador para computador, mas todos eles desempenham as mesmas tarefas. Tomando como exemplo a linha TRS-80 modelo III, tem-se:

```
100 R$ = "Este é um registro do arquivo"
110 OPEN "0",1,"ARQDADOS"
120 PRINT # 1,R$
130 CLOSE 1
```

Esse programa começa com o comando OPEN "0",1, indicativo do sistema operacional que deve estabelecer um canal de comunicação (o canal 1) com a unidade de disco.

O sistema operacional cria arquivos seqüenciais numa série de periféricos diferentes e acessa vários arquivos ao mesmo tempo. Assim, deve-se especificar o canal de comunicação do computador para o periférico e deste para determinado arquivo. O número 1, no caso, indica a unidade de disco.

A seqüência da gravação

A ordem em que os registros são enviados ao arquivo é aquela em que devem permanecer. O único modo de classificar ou editar um arquivo consiste em reorganizá-lo e classificá-lo na memória, e então criar nova versão dele em outra fita.

Após o comando OPEN, vem o nome do arquivo. No TRS-80, este consiste em um nome tal como "ARQDADOS". O método de acesso é "0", indicando que o arquivo é seqüencial e está sendo aberto para gravação (output). Os arquivos seqüenciais podem ser abertos para gravação ou leitura, mas não para ambas ao mesmo tempo. O efeito desse comando é a energização da cabeça de leitura/gravação e a gravação de um marcador de início de arquivo que inclui o nome do arquivo e algumas informações do sistema para assinalar o início do arquivo.

O comando PRINT # 1, na linha 120, envia dados ao arquivo. PRINT aqui não tem o significado habitual de mostrar dados na tela ou na impressora. O sinal # indica que os dados devem ser enviados para o canal especificado, e não enviados para a tela (que é o canal default de saída). O conteúdo da variável R$ é, desse modo, enviado pelo canal 1 ao arquivo seqüencial em disco, chamado "ARQDADOS", onde será gravado. O arquivo agora contém um registro, o string "Este é um registro do arquivo". Close 1 fecha o canal 1 a outras comunicações e grava o arquivo com um marcador de fim de arquivo.

As informações serão lidas posteriormente. Estas são as instruções em BASIC para ler o arquivo que criamos:

```
500 OPEN "1",2,"ARQDADOS"
510 INPUT # 2,A$
520 CLOSE 2
530 PRINT A$
```

Este fragmento de programa, assim como o anterior, usa OPEN. Aqui este comando significa: "Encontre o início do arquivo chamado ARQDADOS e prepare-se para ler seus dados". No programa anterior, ele se traduzia por: "Crie um arquivo chamado ARQDADOS e prepare-se para gravar dados nele". O número de canal indica qual o periférico em uso (no caso, a unidade de disco). O número em si é irrelevante; basta que se abra e se feche um arquivo com o mesmo número de canal.

O nome e o tipo do arquivo são os mesmos porque identificam o arquivo a ser acessado, mas o método de acesso difere — "I" (input, leitura) ao invés de "0" (output, gravação).

Esse programa recebe os dados com a instrução INPUT # 2,A$, significando: "Entrada a partir do canal 2", que é o canal de entrada default. O primeiro registro completo no arquivo é lido e enviado pelo canal 2 para a variável A$, da mesma forma que foi enviado pelo canal 1 a partir da variável R$ por PRINT # 1,R$.

Arquive e encontre

```
10  REM ********* PARA LINHA TRS-80 **********
20  REM *        ARQUIVOS  SEQUENCIAIS        *
30  REM ************************************
40  CLS
50  FOR K=65 TO 90
60  Z$=CHR$(K):C$="GRAVANDO "+CHR$(K)
70  GOSUB 240
80  NEXT K
90  REM ************************************
100 REM *            LER  ARQUIVOS           *
110 REM ************************************
120 FOR L=0 TO 1 STEP 0:FOR M=1 TO 1
130 CLS:PRINT D$;"ACESSO AOS REGISTROS"
140 INPUT "PROCURAR STRING (* = FIM)";N$
150 L$=LEFT$(N$,1):IF L$="*" THEN GOTO 490
160 IF L$<"A" OR L$>"Z" THEN M=0
170 NEXT M
180 Z$=L$
190 GOSUB 370
200 PRINT TAB(5)"REGISTRO ";N;" (USE QQ TECLA)
210 A$=INKEY$:IF A$="" THEN GOTO 210
220 NEXT L
230 END
240 REM ************************************
250 REM *        GRAVAR UM ARQUIVO           *
260 REM ************************************
270 PRINT C$:INPUT "QUANTOS REGISTROS";R
280 OPEN "O",1,Z$
290 IF R=0 THEN PRINT#1,"*":CLOSE 1:RETURN
300 FOR I=1 TO R
310 PRINT C$:PRINT "REGISTRO #";I
320 INPUT "TEXTO...";R$
330 PRINT#1,R$
340 NEXT I
350 PRINT#1,"*":CLOSE 1
360 RETURN
370 REM ************************************
380 REM *          LER UM ARQUIVO            *
390 REM ************************************
400 PRINT "PROCURANDO ";L$;" DE ";N$
410 OPEN "I",1,Z$
420 FOR I=1 TO 100000
430 INPUT#1,R$
440 PRINT R$
450 IF R$="*" THEN N=0:I=100000
460 IF R$=N$ THEN N=1:I=100000
470 NEXT I:CLOSE 1:N$=""
480 RETURN
490 REM ************************************
500 REM *          FIM DO PROGRAMA           *
510 REM ************************************
520 PRINT "FIM DO PROGRAMA":END
```

50-80: Criar arquivos "A" a "Z".

120-180: Entrar com o registro a procurar, encontrar a letra inicial, identificar o arquivo que a contém.

190: Pesquisar o registro dentro do arquivo.

200-210: Mostrar o registro encontrado.

280: Abrir um arquivo para gravação.

290: Se o arquivo estiver vazio, gravar "*"; fechar o arquivo.

320-330: Entrar o texto do registro; gravá-lo no arquivo.

350: Gravar "*" como último registro; fechar o arquivo.

410: Abrir o arquivo para leitura.

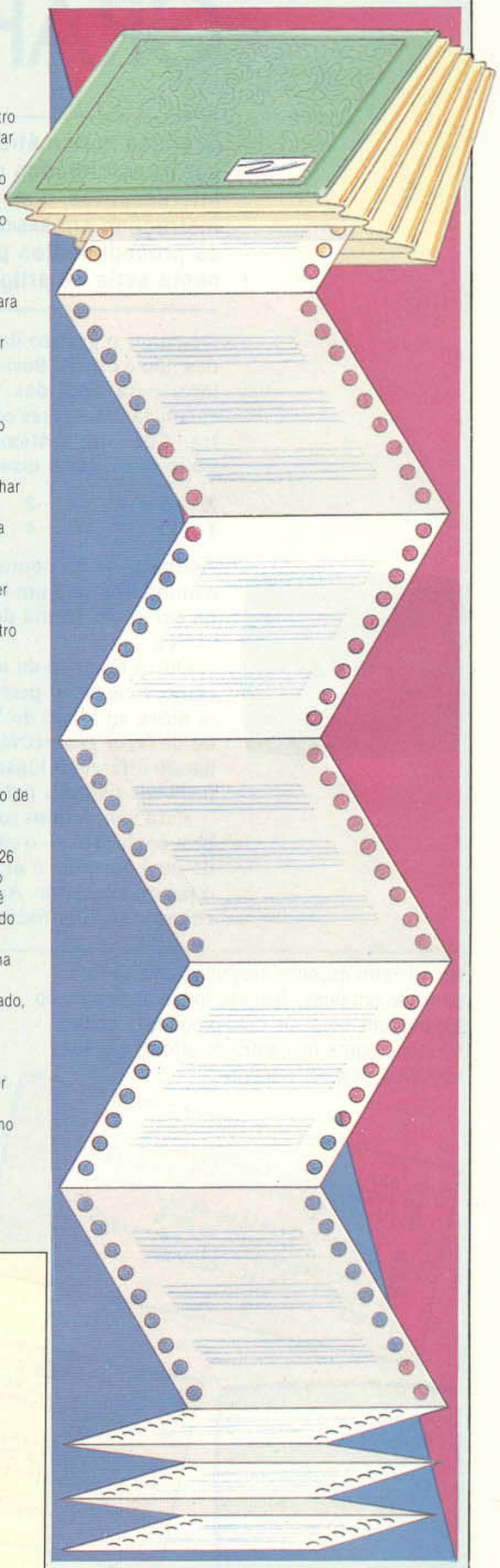
450: Teste de último registro; interromper a pesquisa.

460: Verificar se o registro foi encontrado, interromper a pesquisa.

470: Fechar o arquivo.

Esse programa demonstra o uso de arquivos seqüenciais em disco, criando um índice simples, em ordem alfabética, que abrange 26 arquivos, um para cada letra do alfabeto. Em cada arquivo, você pode digitar registros começando com a letra correspondente. O arquivo pesquisado aparecerá na tela até que o registro seja encontrado; se não for encontrado, a mensagem "REGISTRO" aparecerá na tela.

O programa usa o sistema operacional em disco para obter acesso direto ao arquivo que contém o registro pesquisado; no entanto, o arquivo em si é lido seqüencialmente. Isso reduz o tempo de pesquisa.



Variações

O programa que apresentamos como exemplo trabalha com arquivos em disco, impróprios para a linha Sinclair.

Para trabalhar com arquivos seqüenciais em micros da linha Apple, o processo é um pouco diferente: usa-se o caractere de controle CHR$(4) (que equivale a teclar [CONTROL] D) e as instruções READ e WRITE, além de OPEN e CLOSE.

Assim, o comando para abrir o arquivo seria:

```
PRINT CHR$(4);"OPEN NOMEARQ,S6,D1"
```

Nessa instrução, S6 significa o slot de número 6, e D1 a unidade de disco 1.

Para fechar o mesmo arquivo, usa-se esta instrução:

```
PRINT CHR$(4);"CLOSE NOMEARQ"
```

É dispensável mencionar o slot e o número do drive, na instrução.

Para ler um arquivo, necessitamos destas duas instruções:

```
PRINT CHR$(4);"READ NOMEARQ,REGISTRO"
INPUT R$
```

Para gravar dados no arquivo, as instruções são:

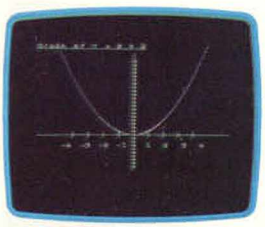
```
PRINT CHR$(4);"WRITE NOMEARQ,REGISTRO"
PRINT R$
```

GRÁFICOS

Gráficos matemáticos podem ser transformados em desenhos interessantes, com algumas simples instruções em BASIC. Veja os procedimentos para obtê-los nesta série de artigos.

Desenhar o gráfico de uma expressão matemática não é difícil. Basta definir uma faixa de valores para uma das variáveis da expressão e encontrar os valores correspondentes para a outra. Com um gráfico simples como $Y = X^2$, pode-se chegar a uma tabela como esta:

X	-5	-4	-3	-2	-1	0	1	2	3	4	5
Y	25	16	9	4	1	0	1	4	9	16	25



Assinalando os pontos em papel quadriculado e unindo-os com um traço, obtém-se a conhecida curva em forma de colar — o gráfico de $Y = X^2$.

Outra maneira de interpretar a curva é vê-la como o caminho percorrido por um ponto que se move ao longo de $Y = X^2$, caminho chamado de *lugar geométrico*. Manipulando e combinando diferentes lugares geométricos, podem-se produzir padrões notáveis com pouco esforço.

Para isso, vamos tomar um lugar geométrico bem conhecido — o círculo. Exprimi-lo por meio de uma fórmula é algo mais complexo do que o método anterior. Ambas as variáveis X e Y da equação são definidas em termos de uma terceira variável ou parâmetro. Variando-se tal parâmetro dentro de um conjunto dado de valores, são encontrados pares de Xs e Ys. E o círculo é representado pela seguinte equação:

X = R SENO(I)
Y = R CO-SENO(I)

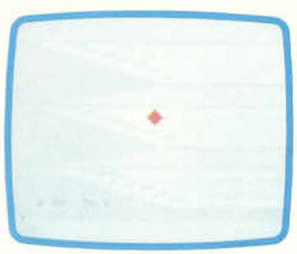
Se determinarmos um conjunto de valores para X e Y enquanto o ângulo I assume valores que completem um círculo (0° a 360°), encontraremos o lugar geométrico de um círculo. O curto programa a seguir foi escrito para o CP-400.

```
10 REM DESENHAR CIRCULO
15 PMODE 4,1:SCREEN 1,0:PCLS
20 XM=256:YM=192:XC=INT(XM/2):YC=INT(XM/2)
30 R=50:PI=3.1416
40 S=PI/20
50 FOR I=0 TO 2*PI STEP S
60 PSET (XC+R*SIN(I),YC+R*COS(I),1)
70 NEXT I
```

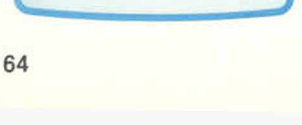
É de notar que, alterando-se o tamanho do STEP entre os pontos demarcados (linha 50), pode-se variar a definição do círculo e a velocidade com que ele é desenhado. A maioria das versões do BASIC não são suficientemente rápidas para desenhar, a velocidade aceitável, círculos uniformes a partir de muitos pontos individuais. Para superar o problema, muitas vezes é preferível usar linhas retas para unir os pontos. De fato, com um grande número de retas chega-se ao meio-termo ideal entre a velocidade e a uniformidade do desenho.

Esse processo também pode ser utilizado para desenhar arcos e elipses. Para arcos, deve-se selecionar diferentes valores de I. Para elipses, dão-se a R valores diferentes a cada uma das fórmulas, de X e de Y. No entanto, o círculo é um fértil ponto de partida para a criação de desenhos interessantes.

No primeiro desenho traçamos uma linha a partir de um ponto fixo até todos os pontos do lugar geométrico. Os dois programas abaixo situam o ponto no centro do círculo e a sua esquerda, respectivamente.



```
10 REM FLOR
15 PMODE 4,1:SCREEN 1,0:PCLS
20 XM=256:YM=192:XC=INT(XM/2):YC=INT
   (YM/2)
30 R=50:PI=3.1416:S=PI/20
40 FOR I=0 TO 2*PI STEP S
50 X=XC+R*SIN(I):Y=YC+R*COS(I)
60 LINE(X,Y)-(XC,YC),PSET
70 NEXT I
```



```
10 REM CIRCULO E PONTO FIXO
15 PMODE 4,1:SCREEN 1,0:PCLS
20 XM=256:YM=192:XC=INT(XM/2):YC=INT
   (YM/2)
30 R=50:PI=3.1416:S=PI/20
40 FOR I=0 TO 2*PI STEP S
50 X=XC+R*SIN(I):Y=YC+R*COS(I)
60 LINE(X,Y)-(40,96),PSET:LINE(XC,YC)-
   (40,96),PSET
70 NEXT I
```

O próximo passo é usar um ponto móvel e não um fixo. Demarcam-se dois lugares geométricos simultaneamente, traçando linhas para ligar pontos correspondentes nas duas rotas. Com o programa abaixo, obtêm-se dois círculos concêntricos ligados por muitas retas.

```
10 REM CIRCULOS CONCENTRICOS
15 PMODE 4,1:SCREEN 1,0:PCLS
20 XM=256:YM=192:XC=INT(XM/2):YC=INT
   (YM/2)
30 R=50:PI=3.1416:S=PI/20
40 FOR I=0 TO 2*PI STEP S
50 X=XC+(R+30)*SIN(I):Y=YC+(R+30)*
   COS(I)
60 P=XC+R*SIN(I):Q=YC+R*COS(I)
70 LINE(X,Y)-(P,Q),PSET
80 NEXT I
```

O programa pode incorporar dados suficientes para desenhar centenas de padrões, mediante mudança de um ou de ambos os lugares geométricos. Uma variação é trocar o seno e o co-seno entre si numa das fórmulas.
Outra consiste em usar potências do seno e do co-seno.

```
10 REM CIRCULOS DESALINHADOS
15 PMODE 4,1:SCREEN 1,0:PCLS
20 XM=256:YM=192:XC=INT(XM/2):YC=INT
   (YM/2)
30 R=50:PI=3.1416:S=PI/20
40 FOR I=0 TO 2*PI STEP S
50 X=XC+(R+30)*COS(I):Y=YC+(R+30)*
   SIN(I)
60 P=XC+R*SIN(I):Q=YC+R*COS(I)
70 LINE(X,Y)-(P,Q),PSET
80 NEXT I
```

Todas essas idéias comportam variações: qualquer livro de matemática fornece fórmulas para obter outras curvas. Mas tente, de preferência, criar seus próprios programas. Pequenas alterações levam o computador a fazer experiências por você.
O último programa aqui mostrado passa por um ciclo infinito de padrões gerados aleatoriamente pelo computador, embora isso não esgote as possibilidades.

```
10 REM CIRCULO E SENO
15 PMODE 4,1:SCREEN 1,0:PCLS
20 XM=256:YM=192:XC=INT(XM/2):YC=INT
   (YM/2)
30 R=50:PI=3.1416:S=PI/20
40 FOR I=0 TO 2*PI STEP S
50 X=XC+SIN(I)*80:Y=YC+(R+30)*SIN(I)
60 P=XC+R*SIN(I):Q=YC+R*COS(I)
70 LINE(X,Y)-(P,Q),PSET
80 NEXT I
```

```
10 REM CIRCULO E SENO*COS
15 PMODE 4,1:SCREEN 1,0:PCLS
20 XM=256:YM=192:XC=INT(XM/2):YC=INT
   (YM/2)
30 R=50:PI=3.1416:S=PI/20
40 FOR I=0 TO 2*PI STEP S
50 X=XC+R*SIN(I)*COS(I):Y=YC+R*SIN(I)*
   COS(I)
60 P=XC+R*SIN(I):Q=YC+R*COS(I)
70 LINE(X,Y)-(P,Q),PSET
80 NEXT I
```

```
10  REM FORMAS CIRCULARES RANDOMICAS
15  PMODE 4,1:SCREEN 1,0:PCLS
20  XM=256:YM=192:XC=INT(XM/2):YC=INT
    (YM/2)
30  R=60:PI=3.1416:S=PI/20
35  PCLS
40  C=RND(50)+1:D=RND(50)+1:E=RND(50)
    +1:F=RND(50)+1
50  FOR I=0 TO 2*PI STEP S
60  X=XC+R*SIN(I/C)*COS(I*D):Y=YC+R*SIN
    (I/E)*COS(I*F)
70  P=XC+R*SIN(I):Q=YC+R*COS(I)
80  LINE(X,Y)-(P,Q),PSET
90  NEXT I
100 A$=INKEY$:IF A$="" THEN GOTO 100
110 GOTO 35
```

Variações

Adaptar e utilizar em outras máquinas as idéias já apresentadas é muito simples. Seu micro deve ser capaz de fazer gráficos de alta resolução (pelo menos 256 x 176, de preferência), ter um BASIC de ponto flutuante com as funções SIN e COS, e comandos para desenhar pontos individuais e linhas retas, tais como o PLOT e o HPLOT do Apple. Os primeiros ajustes necessários são para XM e YM, valores máximos para X e Y que podem ser traçados em seu equipamento. Dependendo das funções que usar, você poderá decidir modificar outras constantes no programa, como R e S. Em seguida, é preciso certificar-se de que seu micro está no modo apropriado para gráficos e selecionar uma cor para o traçado. Finalmente, é necessário um comando para desenhar linhas entre as coordenadas dadas por X e Y, e P e Q. A maioria dos micros possui uma função para traçar linhas absolutas, o que torna simples essa adaptação.
Veja abaixo como os programas FLOR e CIRCULO E SEN * COS foram adaptados para o Apple.

```
10 REM *** FLOR ***
20 HGR2:HCOLOR=7
30 XM=256:YM=192:XC=INT (XM/2):YC=INT
   (YM/2)
40 R=50:PI=3.1416:S=PI/20
50 FOR I=0 TO 2*I STEP S
60 X=XC + R * SIN (I):Y = YC + R *
   COS(I)
70 HPLOT X,Y TO XC,YC
80 NEXT I

10 REM *** CIRCULO E SEN * COS ***
20 HGR2:HCOLOR = 7
30 XM = 256 : YM = 192 : XC = INT (XM/2)
   YC = INT (YM/2)
40 R = 50 : PI = 3.1416: S = PI/20
50 FOR I = 0 TO 2 * PI STEP S
60 X = XC + SIN (I) * COS (I):Y = YC +
   R * SIN (I) * COS (I)
70 P = XC + R * SIN (I):Q = YC + R *
   COS(I)
80 HPLOT X,Y TO P,Q
90 NEXT I
```

Idéias para projetos

1) No loop simples para desenhar um círculo, descrevemos como se criam arcos e elipses a partir do mesmo programa. Agora, tente encontrar uma maneira de traçar espirais.
2) Experimente usar funções como SQR (raiz quadrada) e TAN (tangente) para criar lugares geométricos. Lembre-se, no entanto, de que essas funções devem ser usadas com cuidado, pois tendem a gerar números impraticáveis.
3) Desenvolva versões animadas do programa. Recorrendo a matrizes para registrar as cinco últimas linhas traçadas, pode-se obter um conjunto de cinco linhas que "se perseguem".
4) Que tal criar padrões baseados em três lugares geométricos? Faça dois deles simples (círculo e linha, por exemplo), para evitar que a imagem resulte muito confusa.

MICROS SHARP

Equipados com microprocessador de 8 bits, ROM, RAM, teclado alfabético e conexões para diversos periféricos, o PC-1251 e o PC-1500A, da Sharp, são verdadeiros microcomputadores de bolso.

Sharp PC-1251

O PC-1251, o menor computador existente no mercado, mede 135 x 70 x 10 mm e pesa apenas 115 g. O teclado alfabético tem a metade do tamanho de um teclado comum, o que exige manuseio cuidadoso, pois as teclas têm apenas 4 mm de largura. O PC-1251 apresenta também um teclado numérico comum, com teclas maiores, que permitem usar o micro como calculadora de bolso. O visor de cristal líquido (LCD) mostra uma linha de 24 caracteres por vez, que pode ser movimentada para os lados, e também na vertical, no modo de programação. Mediante um pequeno botão lateral, faz-se o ajuste angular da tela, de modo a adequá-la às condições de iluminação reinantes.

O PC-1251 funciona em três modos, acionados por uma chave seletora: modo de programação (PRO); de atribuição de funções a determinadas teclas (RSV); e modo RUN, para executar os programas ou utilizar a calculadora. Para armazenar e imprimir programas e dados, o PC-1251 conta com uma unidade à parte, a impressora/microcassete CE-125, na qual se encaixa o microcomputador.

A versão do BASIC, bastante eficiente considerando-se o tamanho da máquina, contém funções úteis, como ASC e CHR$, embora não possua a opção ELSE para o comando IF-THEN. Outra desvantagem é a utilização dos mesmos nomes, de uma letra, para as variáveis numéricas e os strings: se A for usado para um número, A$ não servirá para o string. As mensagens de erro reduzem-se a nove e expressam-se apenas como uma letra, o que dificulta a depuração dos programas. Em cada programa, a numeração das linhas vai até 999. Para utilizá-lo, é necessário digitá-lo novamente, a menos que se disponha de uma unidade de microcassete adicional. A memória de 4 Kbytes comporta um programa por vez, e mesmo dividindo-os em subprogramas separados por END e executados por um GOTO, esse recurso só serve para programas extremamente reduzidos.

Sharp PC-1500A

Três vezes mais pesado (375 g) e com dimensões maiores, o PC-1500A é superior ao PC-1251 em termos de teclado, BASIC e possibilidades de conexão a periféricos. Seu visor LCD mostra 26 caracteres, sem ajuste angular para melhor visibilidade. Este micro dispõe de seis teclas adicionais programáveis para até dezoito funções. As teclas alfabéticas da fileira superior já vêm programadas com comandos BASIC: as palavras-chave não estão escritas nas teclas, mas indicadas numa máscara plástica.

O BASIC Microsoft deste modelo é superior ao do PC-1251. As variáveis podem ter duas letras e o mesmo nome serve para variáveis numéricas e strings; conta ainda com recursos gráficos e sonoros, além de comandos ON ERROR GOTO, TRON e TROFF e um relógio-calendário residente, de funcionamento permanente. Há 39 mensagens de erro para a máquina padrão e outras dezesseis usadas com os periféricos adicionais — estas expressam-se apenas sob a forma de números, o que dificulta a depuração.

A RAM de 8,5 Kbytes pode ser expandida de várias maneiras. Na parte inferior da máquina encaixa-se um cartucho de expansão de memória, oferecido em dois tipos: volátil (só mantém os dados enquanto acoplado à máquina), com 4 ou 8 Kbytes; e não volátil (acionado a pilha), com 8 ou 16 Kbytes.

Outro recurso adicional do PC-1500A, disponível no mercado, é uma interface dotada de saída paralela e serial para comunicação com impressoras e computadores até de grande porte. Encontra-se também no mercado o periférico "tablete de software" (Software Board), que contém 140 teclas tipo membrana, programáveis pelo usuário para executar diferentes tarefas, como totalizar colunas de números e chamar subrotinas.

Com seu BASIC bastante poderoso e recursos gráficos e de comunicação, este micro de bolso da Sharp ressente-se apenas da ausência de software comercialmente disponível. O fabricante oferece uma fita com uma seleção de programas. Para utilização com impressora, o PC-1500A conta com uma série de comandos BASIC, que controlam a produção e a impressão de imagens.

O micro é acompanhado de um livro com listagens de 53 programas, que permitem aplicações em diversas áreas: matemática, estatística, eletricidade, comércio e jogos.

Compartimento para pilhas
Contém as quatro pilhas necessárias para acionar o computador.

Microprocessador
Projetado para o PC-1500, com endereçamento em 8 bits.

Saída para cartucho
Interface que permite o acoplamento de cartuchos de RAM adicional. Acopla-se aqui também a unidade impressora-cassete, ou uma interface serial.

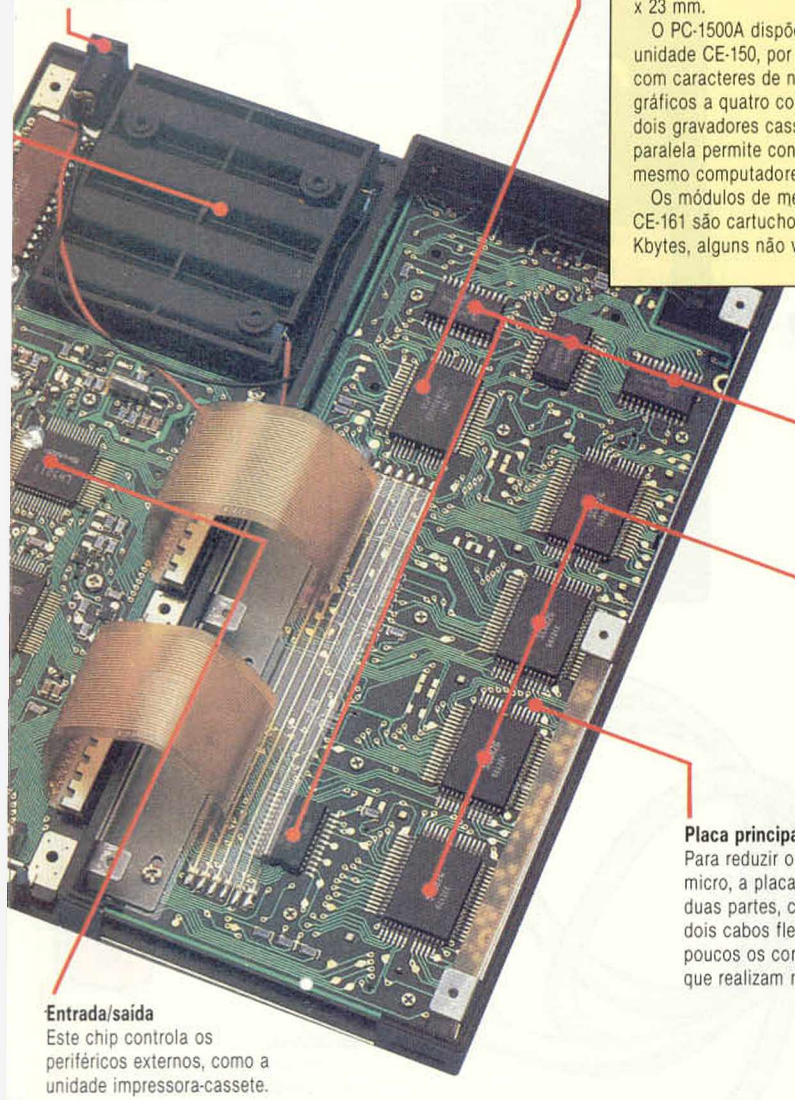
Entrada para alimentação externa
O computador também é acionado por corrente alternada, usando-se um transformador.

ROM com BASIC
Este chip contém o BASIC Microsoft utilizado no computador.

Periféricos Sharp
O Sharp modelo PC-1251 encaixa-se na unidade adicional CE-125, dotada de gravador de microfita e impressora térmica, que imprime 24 caracteres por linha. A unidade mede 205 x 149 x 23 mm.

O PC-1500A dispõe de grande variedade de periféricos. A unidade CE-150, por exemplo, contém uma impressora-plotter com caracteres de nove tamanhos, recursos para imprimir gráficos a quatro cores e cartesianos. Conta com interface para dois gravadores cassete. A interface CE-158 padrão RS232C e paralela permite conexão com quase todos os tipos de micro e mesmo computadores de grande porte.

Os módulos de memória adicional CE-151, CE-155, CE-159 e CE-161 são cartuchos contendo RAM do tipo CMOS, de 2 a 16 Kbytes, alguns não voláteis.



Chips de RAM
Fornecem 8,5 K de memória, dos quais 6,6 são disponíveis para o usuário.

Chips de tela
Estes quatro chips organizam os dados para exibição no visor de cristal líquido.

Placa principal
Para reduzir o tamanho do micro, a placa foi dividida em duas partes, conectadas por dois cabos flexíveis. São poucos os componentes CMOS, que realizam múltiplas funções.

Entrada/saída
Este chip controla os periféricos externos, como a unidade impressora-cassete.

SHARP PC-1251

MEMÓRIA
4 K de RAM.

TELA
Visor LCD, 1 x 24 caracteres.

TECLADO
Alfabético e numérico.

LINGUAGENS
BASIC simples.

PERIFÉRICOS
Unidade com microfita e impressora térmica com 1 x 24 caracteres.

SHARP PC-1500A

MEMÓRIA
8,5 K de RAM, expansível por cartuchos.

TELA
Visor LCD, 1 x 26 caracteres.

TECLADO
Alfabético e numérico, com seis teclas programáveis para dezoito funções.

LINGUAGENS
BASIC com muitos recursos.

PERIFÉRICOS
Unidade com impressora-plotter a quatro cores e saída para dois gravadores cassete. Interface para comunicação com outros micros.

Sharp PC-1251

Pesando apenas 115 g, esta calculadora com recursos de micro é útil para engenheiros e cientistas. O pequeno tamanho das teclas, no entanto, não é adequado para as tarefas que exigem muita digitação de texto.

Sharp PC-1500A

Este micro de bolso conta com um BASIC poderoso e muitos recursos úteis, como relógio-calendário permanente, impressora a quatro cores e saída para comunicação com outros micros e computadores de grande porte.

FERRAMENTAS ESSENCIAIS

Pequenos consertos podem ser feitos em casa. O domínio das técnicas de soldagem, por exemplo, muitas vezes evita que o principiante recorra aos dispendiosos serviços especializados em micros.

O processo de soldagem consiste em juntar dois objetos de metal por meio de uma liga metálica branda — que se funde com facilidade. Para trabalho em eletrônica, emprega-se uma liga de chumbo e estanho.

De início, os objetos devem ser aquecidos, por meio do aparelho de solda, a temperatura superior ao ponto de fusão da liga (280°C).

Em seguida, aplica-se a solda ao componente (e não à ponta do soldador) e retira-se o aparelho. Ao esfriar, a solda solidifica e a junção está feita.

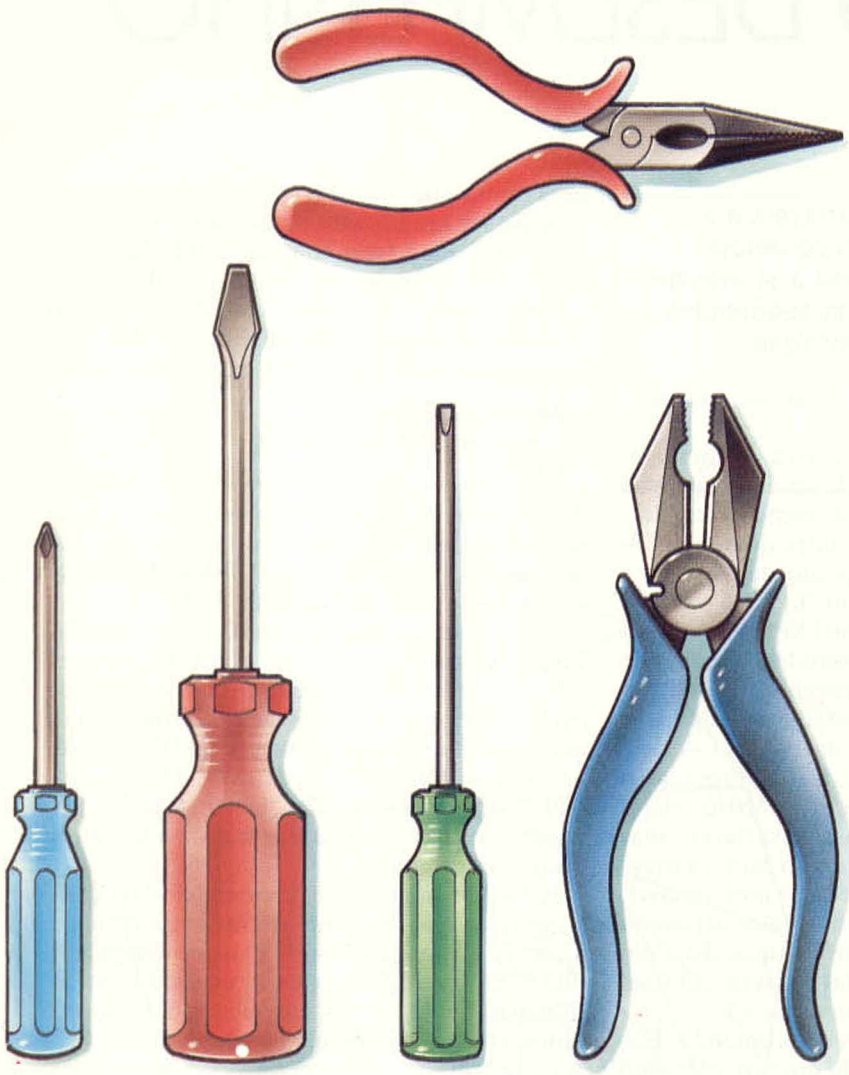
Quando se aplica o soldador sobre o fio de solda, a liga derrete com muita rapidez. Contudo, ela retorna ao estado sólido assim que entra em contato com o componente frio. O resultado é insatisfatório. No melhor dos casos, não haverá junção; no pior, a ligação ficará muito fraca, impedindo o fluxo contínuo de eletricidade. A única forma de evitar que isso aconteça é aquecer o componente até que a solda derreta.

Ferramentas de bancada

Para trabalhar com os minúsculos componentes dos computadores, encontra-se à venda uma série de tornos, pinças e prendedores, a custo relativamente baixo. Se, com tal equipamento, o aperto for ainda insuficiente, tente fazer calços usando fita adesiva, com o lado aderente para fora, de modo a fixar melhor o componente na pinça.

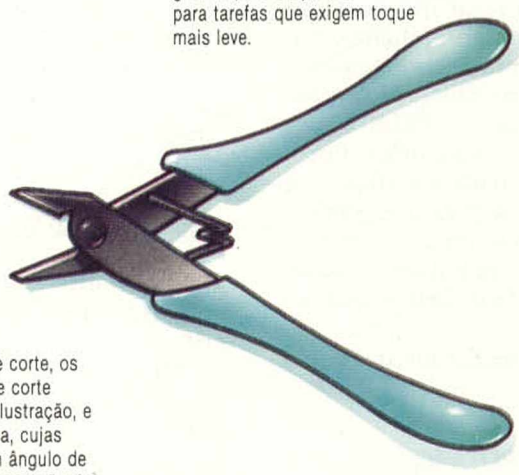
Soldadores
Os componentes eletrônicos são, na maioria, sensíveis ao aquecimento excessivo. Daí a importância da escolha correta do soldador, do tamanho de sua ponta e do diâmetro da solda usada. Para soldar cabos, fios e conectores, basta um soldador de 15 a 25 W e solda de uso geral de 1,5 mm. Por ser demorado seu aquecimento, esses aparelhos permitem trabalhar com segurança em componentes delicados.

Descascador de fios
Sempre que usar cabos ou fios, é preciso descascar com cuidado toda a cobertura e o isolante. Descascadores de fios simples, como o da figura, custam pouco e podem ser pré-ajustados a várias bitolas de fio, de modo a remover o isolante, deixando o fio intato.



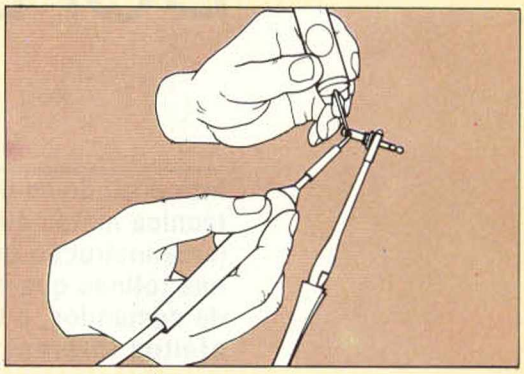
Chaves adequadas
Há dois tipos de chave de parafuso: de fenda e em cruzeta. Esta última pode ter ponta Philips ou Pozidriv. Na maioria dos micros, serve qualquer um dos dois tipos. Convém, no entanto, dispor de ambos em vários tamanhos.

Alicates
O principiante precisa de apenas dois tipos de alicate — o de bico chato, para serviços pesados, como cortar fios grossos, e o de ponta, para tarefas que exigem toque mais leve.



Alicate de corte
Entre os alicates de corte, os principais são os de corte lateral, como o da ilustração, e os de corte de ponta, cujas lâminas formam um ângulo de 90° com o plano de rotação da ferramenta. Numa emergência, porém, um cortador de pregos pode resolver o problema.

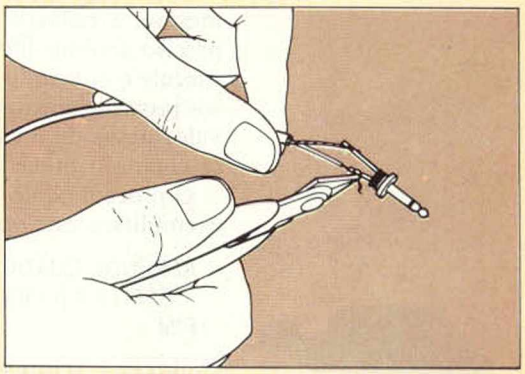
Soldagem de plugue



Descascar e estanhar
O primeiro estágio do processo de soldagem de um fio a um plugue consiste em descascar os fios do cabo, removendo o isolante. Retire mais fio que o necessário e, depois, corte no tamanho certo. Fixe o cabo no torno ou na pinça e aqueça-o com o soldador. Aplique a solda ao cabo e, quando começar a derreter, use o soldador para recobrir com a solda toda a parte exposta do cabo. Esse processo, chamado estanhamento, torna mais simples a soldagem — nesse ponto, a solda está no componente.

Repita o mesmo procedimento no plugue. Enquanto o terminal está quente o bastante para derreter a solda, encoste o cabo estanhado e só retire o soldador quando ambas as soldas (do componente e do cabo) estiverem fundindo. Depois, sopre-as para resfriar mais depressa e está pronta a junção.

Acabamento
O último estágio do trabalho consiste em remover as sobras de material. Corte os cabos bem curtos, assim como os pinos terminais nos componentes — mas antes teste o funcionamento. A razão para cortar é simples: pontas longas sobrando no cabo ou pinos do terminal protuberantes podem entrar em contato com outra peça e causar curto-circuito ou, pior, um desses irritantes defeitos de ocorrência intermitente.



DUPLO DESEMPENHO

Introduzindo-se dados variáveis e a técnica matemática da recorrência (uma instrução que chama a si mesma) nas rotinas que executam seqüências de comandos, o LOGO consegue efeitos interessantes.

RECURSION

Por este termo, traduzido como "recorrência", indica-se uma série de rotinas repetidas dentro de um programa, na qual o resultado de cada repetição depende do resultado da repetição anterior. A rotina chama a si própria até a tarefa estar completa.

Na linguagem LOGO, uma palavra — como TAMANHO, por exemplo — pode ser empregada de três maneiras distintas. A diferença entre tais usos é assinalada pelas seguintes notações: TAMANHO, :TAMANHO ("dois pontos tamanho") e "TAMANHO ("aspas tamanho").

Ao encontrar a palavra TAMANHO sem nenhuma pontuação, o LOGO a considera o nome de uma rotina e executa os comandos da definição de TAMANHO. :TAMANHO é usada para designar o valor atribuído ao nome da variável — se o LOGO encontrar :TAMANHO, ele irá procurar o valor associado a esse nome. "TAMANHO refere-se aos procedimentos e nomes de variáveis, mas indica apenas seus nomes, e não seus valores.

Portanto, "TAMANHO refere-se a uma variável, enquanto :TAMANHO designa o valor atribuído a essa variável. (Observe que as aspas são colocadas antes, e não depois das palavras. O final do nome é indicado por um espaço.)

Na versão MLOGO, depois dos comandos EDITE, SEM e LISTAR, o nome de um procedimento é escrito sem aspas. Desse modo, a sintaxe correta é EDITE QUADRADO, mesmo que QUADRADO não seja uma chamada para o procedimento QUADRADO, mas apenas o nome desse procedimento.

Quando você quiser empregar qualquer dos procedimentos definidos até agora, digite o nome do procedimento, como faria com qualquer comando, como DESENHE ou SEMT. No entanto, o uso de outros comandos — FRENTE, por exemplo — exige informações adicionais. Por si mesma, a palavra FRENTE nada significa — é preciso atribuir-lhe um valor para que o LOGO execute o comando. Incluindo variáveis em nossos procedimentos, podemos introduzir qualquer valor desejado e variar o efeito obtido quando se chama o procedimento.

O procedimento para desenhar um quadrado exemplifica essa variação:

```
APRENDA QUADRADO
  REPITA 4 [FRENTE 50 DIREITA 90]
FIM
```

Como está, o procedimento traça um quadrado cujos lados medem 50 unidades. Contudo, ele seria muito mais útil se o tamanho do quadrado pudesse ser modificado de acordo com valores escolhidos. Para fazer com que QUADRADO aceite esse tipo de dado, use os recursos de edição, a fim de substituir o valor fixo 50 pela variável "LADO e acrescente :LADO à linha do título. Com as modificações, o procedimento ficará assim:

```
AP QUADRADO :LADO
  REPITA 4 [FR :LADO DI 90]
FIM
```

Agora, quando chamar um procedimento, você precisa atribuir um valor à variável "LADO. Experimente QUADRADO 40, QUADRADO 10 etc. para ver como varia o tamanho da figura.

Veja o que acontece quando se digita QUADRADO 30. A primeira coisa que o LOGO faz é consultar a definição de QUADRADO. A linha do título informa que é necessária a introdução de um dado, e que ele se chamará "LADO. O valor na linha de entrada — nesse caso, 30 — é, portanto, atribuído à variável "LADO. A seguir, são obedecidos os comandos incluídos na definição do procedimento.

A melhor maneira de se entender o processo é imaginar que os nomes de variáveis remetem a caixas contendo valores. Quando chega à linha FRENTE :LADO, o LOGO procura a caixa intitulada "LADO, lê o valor ali encontrado e o utiliza como um dado para o comando FRENTE. A caixa "LADO é usada apenas nesse procedimento. Se algum outro também utilizar "LADO como nome de uma variável, usará outra caixa. Portanto, "LADO é uma variável *local*.

Os dados variáveis também podem ser utilizados em subprocedimentos. O procedimento CASA é uma das soluções do exercício proposto no artigo anterior. Ele pode ser modificado de modo a aceitar valores diferentes:

```
AP CASA :GRANDE
  QUADRADO :GRANDE
  FR :GRANDE DI 30
  TRIANGULO :GRANDE
  ESQUERDA 30 VOLTE :GRANDE
FIM
AP QUADRADO :TAMANHO
  REPITA 4 [FR :TAMANHO DI 90]
FIM
AP TRI :LADO
  REPITA 3 [FR :LADO DI 120]
FIM
```

Três nomes diferentes de variáveis constam desses procedimentos — "GRANDE, "TAMANHO e "LADO. Pode-se usar o mesmo nome para as três variáveis, já que elas se restringem aos procedi-

mentos nos quais são empregadas. Mas isso dá margem a muita confusão.

Para entender como funciona esse procedimento, examine o que acontece quando se digita CASA 30. O LOGO lê a linha de entrada e atribui o valor 30 à variável "GRANDE, em CASA. Assim, a primeira linha de CASA equivale agora a QUADRADO 30. À variável "TAMANHO, em QUADRADO, atribui-se o valor 30. Quando QUADRADO é executado, FRENTE :TAMANHO torna-se FRENTE 30. Procedimento semelhante é seguido no caso de TRI.

Experimente adaptar os procedimentos para o traçado de um tabuleiro de 5 x 5 quadrados, de modo que o tamanho do quadrado seja um dado de entrada.

A seguir, apresentamos um procedimento para o desenho de polígonos no qual o número de lados varia:

```
AP POLI :LADOS
  REPITA :LADOS [FR 50 DI 360/ :LADOS]
FIM
```

Nesse procedimento com uma variável, POLI 3 traçará um triângulo, POLI 4 um quadrado, e assim por diante. Contudo, em todos os polígonos, os lados terão 50 unidades de comprimento.

Um procedimento mais geral, que trace polígonos de qualquer tamanho, requer duas variáveis — uma para o número de lados e outra para o comprimento dos lados. Adapte o procedimento POLI de modo que 50 seja substituído por um nome de variável, acrescentado à linha do título.

```
AP POLI :LADOS :TAMANHO
  REPITA :LADOS [FR :TAMANHO DI 360/ :LADOS]
FIM
```

Assim, POLI 5 30 traçará um pentágono, cujos lados medem 30 unidades. De modo semelhante, você pode modificar sua nova versão do procedimento do tabuleiro, para que ele desenhe tabuleiros de qualquer tamanho (e não apenas de 5 x 5 quadrados). Empregue duas variáveis — o número de quadrados em cada direção (horizontal e vertical) e o tamanho deles.

Variáveis globais

Além das variáveis locais em relação aos procedimentos que as utilizam, é possível definir variáveis que sejam usadas por todos os procedimentos. Conhecidas como "globais", são úteis para se estabelecer comunicação entre procedimentos diferentes. Mas seu emprego apresenta a desvantagem de tornar mais difícil a depuração de erros e, por isso, elas devem ser usadas com critério.

Usa-se o comando FACA para atribuir valores às variáveis globais. FACA "LADO 3 significa que o valor 3 está sendo associado à variável "LADO. FACA "LADO :LADO + 1 aumenta em 1 o valor de "LADO. Nesse exemplo, o significado preciso da notação é: encontre o valor da variável "LADO, adicione 1 e atribua o resultado à própria variável "LADO. Em ambos os casos, FACA exige a entrada de dois dados — o nome da variável e o valor que será associado a ela.

Como resumo dos recursos de programação tratados neste artigo, criamos alguns procedimentos para o traçado de espirais. O procedimento principal receberá o nome de ESPIRAL. Ele opera com três variáveis: o comprimento inicial da linha a ser traçada, o ângulo em que a tartaruga deve girar a cada rotação da espiral e um fator de escala pelo qual o comprimento inicial deve ser multiplicado (para se produzir o efeito de espiral). Conjuntos diferentes de valores podem ser usados para se obter efeitos diferentes — nós experimentamos com 70 283 0.95, 70 143 0.95 e 20 243 1.05. Tente outros conjuntos de números e veja o que acontece.

FECHADO é um novo comando. Ele interrompe a tartaruga e emite uma mensagem de erro, quando ela transpõe as margens da tela. Em muitos casos, a ultrapassagem dos limites da tela produz resultados interessantes. No entanto, a rotina que estamos estudando exige o uso de FECHADO, pois a ultrapassagem das margens da tela estraga o efeito de espiral.

O procedimento principal ESPIRAL desenha repetidas vezes uma linha, cujo comprimento é determinado pelo fator escalar. A seguir, ele gira

Traçado da espiral

```
AP E.FRENTE :NUMERO
  FRENTE :ESCALA
FIM
AP ESPIRAL :LADO :ANGULO :FATOR
  INICIO
  REPITA 1000 [E.FRENTE :LADO DI :ANGULO
    AUMENTO :FATOR]
FIM
AP INICIO
  DESENHE FECHADO FACA "ESCALA :LADO
FIM
AP AUMENTO :DIST
  FACA "ESCALA :ESCALA * :DIST
FIM
```

Exercício 3

A) Escreva um procedimento para traçar um círculo de raio 50. Modifique o procedimento de modo que o raio possa ser variável.
B) Escreva o procedimento para o traçado de um "alvo" (cinco círculos concêntricos).

Registre

Linha Sinclair

Como usar muitas vezes a mesma função dentro de um programa, sem redigitá-la? Defina a função como um string relacionado a uma variável alfanumérica:

```
LET A$ = "X**2+Y*3-1"
```

Depois, para utilizar a função, basta pedir o valor da variável:

```
PRINT VAL A$
```

Esse recurso pode ser adaptado sem dificuldade para qualquer outro tipo de microcomputador.

Respostas do exercício 2

A) Quebra-cabeça Tangram

O homem sentado, o homem rezando e o gato usam o lado do paralelogramo oposto ao lado usado pelo cão no problema anterior. Em LOGO, para inverter uma forma, basta transformar os comandos DIREITA em ESQUERDA, e vice-versa. Assim, em vez de formar o paralelogramo como:

```
AP PAR
  REPITA 2 [FR 25 DI 45 FR 35 DI 135]
```

nós usaríamos um paralelogramo assim:

```
AP PAR1
  REPITA 2 [FR 25 ES 45 FR 35 ES 135]
```

Homem correndo

```
AP CORRENDO
 MOVE1 TRI1 MOVE2 PAR MOVE3 TRI3 MOVE4
 TRI3 MOVE5 QUADRADO MOVE6 TRI1 MOVE7
 TRI2 MOVE8
FIM
AP MOVE1
 ES 45
FIM
AP MOVE2
 SC FR 25 DI 135 FR 17.5 ES 45 CL
FIM
AP MOVE3
 SC FR 75 DI 90 CL
FIM
AP MOVE4
 SC DI 90 FR 25 DI 90 CL
FIM
AP MOVE5
 SC FR 50 DI 135 FR 50 DI 135 CL
FIM
AP MOVE6
 SC DI 135 FR 21 DI 135 FR 25 ES 90
 FR 50 ES 90 FR 25 DI 90 CL
FIM
AP MOVE7
 SC FR 25 DI 135 FR 71 DI 45 VO 35 CL
 FIM
AP MOVE8
 SC FR 35 ES 90 FR 25 DI 45 FR 17.5
 ES 45 FR 25 DI 135 CL
FIM
```

Homem sentado

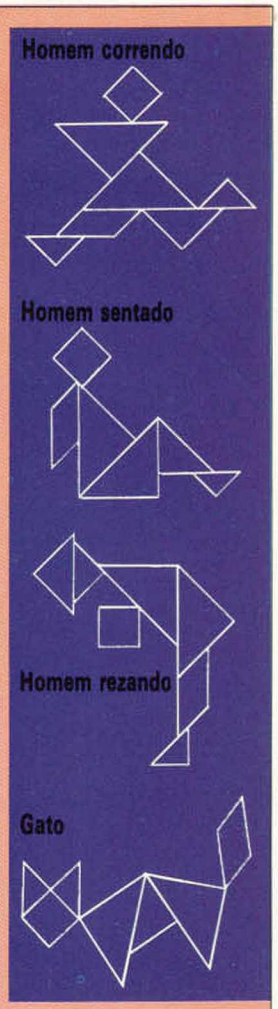
```
AP SENTADO
 MOVE1 TRI1 MOVE2 TRI2 MOVE3 TRI3 MOVE4
 TRI1 PAR1 MOVE5 QUADRADO MOVE6 TRI3
 MOVE7
FIM
AP MOVE1
 ES 45
FIM
AP MOVE2
 SC FR 25 ES 45 FR 17.5 DI 90 CL
FIM
AP MOVE3
 SC VO 15 ES 90 CL
FIM
AP MOVE4
 SC FR 50 DI 45 FR 25 DI 90 CL
FIM
AP MOVE5
 SC FR 25 ES 45 FR 35 ES 45 CL
FIM
AP MOVE6
 SC VO 50 ES 90 CL
FIM
AP MOVE7
 SC VO DI 135 VO 50 DI 90 FR 35
 ES 90 CL
AP MOVE8
 SC FR 35 ES 90 FR 25 DI 45 FR 17.5
 ES 45 FR 25 DI 135 CL
FIM
```

Homem rezando

```
AP REZANDO
 MOVE1 TRI1 MOVE2 PAR1 MOVE3 TRI3 MOVE4
 TRI3 MOVE5 TRI1 MOVE6 TRI2 MOVE7
 QUADRADO MOVE8
FIM
AP MOVE1
 ES 90
FIM
AP MOVE2
 SC FR 25 DI 135 FR 30 CL
FIM
AP MOVE3
 SC ES 45 FR 35 ES 135 VO 50 CL
FIM
AP MOVE4
 SC DI 90 FR 50 ES 135 CL
FIM
AP MOVE5
 SC DI 90 FR 50 ES 135 FR 5 DI 90 CL
FIM
AP MOVE6
 SC DI 90 FR 25 DI 45 FR 7.5 DI 45
 VO 35 CL
FIM
AP MOVE7
 SC FR 35 DI 135 FR 7.5 ES 45 FR 55
 DI 45 CL
FIM
AP MOVE8
 SC ES 45 FR 36 DI 45 FR 56 ES 135
 FR 5 ES 45 VO 25 CL
```

Gato

```
AP GATO
 MOVE1 TRI3 MOVE2 QUADRADO MOVE3 TRI1
 MOVE4 TRI1 MOVE5 TRI3 MOVE6 PAR1
 MOVE7 TRI2 MOVE8
FIM
AP MOVE1
 SC FR 50 DI 90 CL
FIM
AP MOVE2
 DI 170
FIM
AP MOVE3
 SC DI 90 FR 25 ES 90 CL
FIM
AP MOVE4
 DI 180
FIM
AP MOVE5
 SC DI 90 FR 25 ES 80 FR 50 DI 45
 FR 50 DI 90 CL
FIM
AP MOVE6
 ES 155
FIM
AP MOVE7
 SC ES 160 FR 35 CL
FIM
AP MOVE8
 SC FR 35 ES 45 FR 21 DI 135 CL
FIM
```



B) Casa

```
AP CASA
 QUADRADO FR 50 DI 30
 TRI ES 30 VO 50
FIM

AP QUADRADO
 REPITA 4 [FR 50 DI
 90]
FIM

AP TRI
 REPITA 3 [FR 50 DI
 120]
FIM
```

C) Quadro

```
AP QUADRO
 REPITA 5 [LINHA FR
  10] VO 50
FIM

AP LINHA
 REPITA 5[ QUADRADO DI
 90 FR 10 ES 90] ES 90
 FR 50 DI 90
FIM

AP QUADRADO
 REPITA 4 [FR 10 DI 90]
FIM
```

D) Estrela de seis pontas

```
AP SEIS.ESTRELAS
 TRI MOVE TRI
 MOVE.VOLTA
FIM

AP TRI
 DI 30 REPITA 3
 [FR 50 DI 120]
FIM

AP MOVE
 SC FR 29 DI 60 CL
FIM

AP MOVE.VOLTA
 SC ES 60 VO 29 CL
FIM
```

a tartaruga num ângulo fixo; e, por fim, altera o fator escalar.

O comprimento das linhas traçadas aumenta ou diminui conforme o fator escalar seja maior ou menor do que 1. O número elevado depois de REPITA serve apenas para manter o procedimento rodando por bastante tempo. Se você acha que já viu suficientemente o procedimento, acione [CONTROL]-G para interrompê-lo.

Todas as variáveis são locais, com a exceção de "ESCALA. Ela é uma variável global porque "AUMENTO muda seu valor, e este novo valor deve estar disponível para E.FRENTE. "ESCALA serve, portanto, para a comunicação entre dois procedimentos.

Recurso à recorrência

Um dos primeiros programas definidos nesta seção foi o procedimento para o desenho de um quadrado. Comandada por ele, a tartaruga movia-se para a frente um certo número de unidades, girava 90° para a direita, e repetia três vezes essas operações. Apresentamos em seguida outra maneira de se desenhar um quadrado:

```
AP QUADRADO
  FR 50
  DI 90
  QUADRADO
FIM
```

Se você testar esse procedimento, verá que a tartaruga traça um quadrado e depois continua a se movimentar pelo perímetro da figura, até que se tecle [CONTROL]-G. A característica mais evidente desse novo procedimento QUADRADO é que ele "chama" a si mesmo — em outros termos, ele é recorrente (ou recursivo).

Quando se roda esse procedimento, o LOGO consulta a definição de QUADRADO e passa a obedecer às instruções. A tartaruga move-se para a FRENTE 50 unidades e depois gira 90° para a DIREITA.

A instrução seguinte é QUADRADO. O LOGO volta portanto a consultar a definição e começa a segui-la. Isso continuará se repetindo indefinidamente se o programa não for interrompido.

As chamadas recorrentes também são usadas em procedimentos que requerem dados:

```
AP POLI :LADO :ANGULO
  FR :LADO
  DI :ANGULO
  POLI :LADO :ANGULO
FIM
```

Esse procedimento traça inúmeros polígonos. Como exercício, você poderia utilizar no procedimento um ângulo de 89°. Outra possibilidade é a mudança do valor do dado na chamada recursiva. Assim:

```
AP POLIESPI :LADO :ANGULO
   FR :LADO
   DI :ANGULO
   POLIESPI (:LADO + 5) :ANGULO
FIM
```

A única diferença entre este procedimento e o anterior é que há um acréscimo de 5 unidades ao valor de LADO cada vez que ele é chamado. Portanto, se você começar com POLIESPI 10 90, a primeira chamada traçará uma linha de 10 unidades de comprimento; a segunda traçará um linha de 15; a terceira, de 20; e assim por diante. O resultado será uma espiral. Experimente com dados diferentes: comece com 10 90, 10 95, 10 120, 10 117, 10 144 e 10 142. Você também pode modificar o procedimento transformando a adição em subtração ou multiplicação.

Mostraremos agora um procedimento que aumenta o ângulo, em vez de aumentar o valor do lado:

```
AP ESPIRAL :LADO :ANGULO :AUM
  FR :LADO
  DI :ANGULO
  ESPIRAL :LADO (:ANGULO + :AUM) :AUM
FIM
```

Experimente com diversos valores: 5 0 7, 10 40 30, 15 2 20, 5 30 20. Por que algumas figuras são fechadas e outras não? Você poderia encontrar uma regra para isso?

A repetição simples de um trecho de código é conhecida como "iteração". O LOGO usa REPITA com essa finalidade, enquanto outras linguagens empregam outros comandos, tais como FOR-NEXT, REPEAT-UNTIL e WHILE-WEND. Se você já programou em outras linguagens, talvez sinta dificuldade para romper o hábito de utilizar a iteração. Mas valerá a pena, pois os grafismos do LOGO são ideais para a experimentação com as chamadas recorrentes.

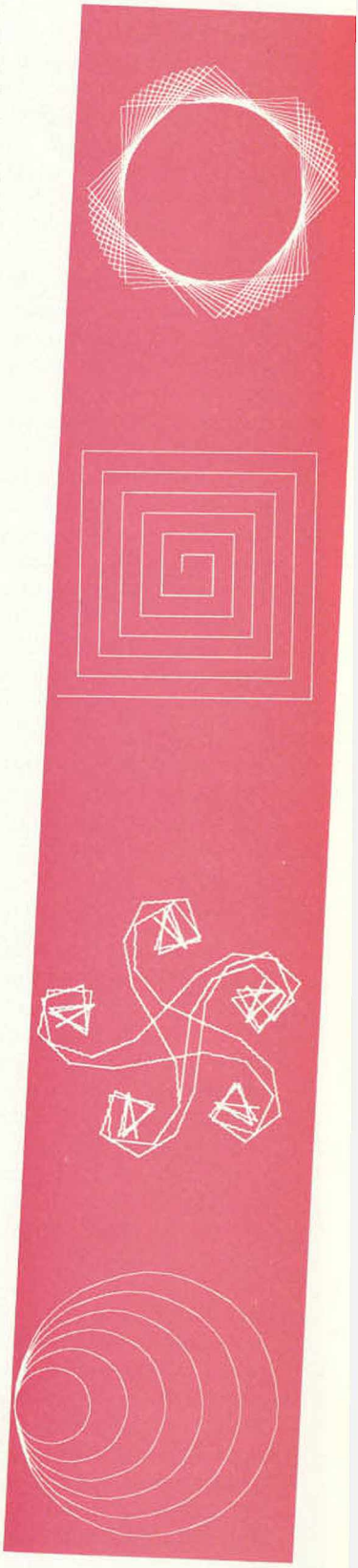
Regras de interrupção

Os procedimentos recursivos citados continuam a se repetir indefinidamente. Assim, precisamos dispor de um meio para interrompê-los em dado instante. Tomando como exemplo o procedimento QUADRADO, poderíamos interrompê-lo logo após o traçado do primeiro quadrado completo, quando a direção da tartaruga estiver de volta em 0. Isso pode ser feito acrescentando-se uma "regra de interrupção" ao procedimento:

```
AP QUADRADO :LADO
  FR :LADO
  DI 90
  SE DIRECAO = 0 ENTAO PARE
  QUADRADO :LADO
FIM
```

Os novos comandos primitivos são PARA e SE. O primeiro interrompe a execução de um procedimento e devolve o controle para o procedimento que o chamou. A declaração SE é o modo pelo qual o LOGO toma decisões. SE é seguido por uma condição, e ENTAO, por uma ação realizada *se* a condição for verdadeira.

Vamos examinar uma versão de POLIESPI e ver em detalhes o que ocorre quando nela se introduz uma regra de interrupção:



```
AP POLIESPI :COMPRIMENTO
  SE :COMPRIMENTO > 15 ENTAO PARE
  FR :COMPRIMENTO
  DI :90
  POLIESPI (:COMPRIMENTO – 5)
FIM
```

Isso é o que acontece quando rodamos POLIESPI 10. O procedimento POLIESPI é chamado e associa-se o valor 10 a uma variável local. Como tal valor não supera 15, o LOGO passa a executar o movimento FRENTE 10 DIREITA 90. Depois, chama POLIESPI de novo, mas desta vez o valor da variável é 15, o que leva a nova chamada de uma cópia do procedimento. Como COMPRIMENTO não ultrapassa 15, a tartaruga é movida para a FRENTE 15 DIREITA 90 e, então, realiza-se nova chamada para POLIESPI.

Contudo, a variável local foi aumentada para 20, de modo que o procedimento pára e devolve o controle ao procedimento que o chamou (POLIESPI 15). Este, por sua vez, ao chegar ao final, devolve o controle a seu procedimento de chamada. Quando este termina, o programa se encerra.

A recorrência no LOGO opera com procedimentos que chamam cópias de si mesmos. As chamadas recursivas são cópias que existem lado a lado com o procedimento original, embora funcionem como se fossem diferentes deste.

Quando chegam ao final, essas cópias sempre transferem o controle ao procedimento que as chamou. Para ilustrar esse processo de retorno a partir das chamadas de procedimento, podemos reescrever POLIESPI do seguinte modo:

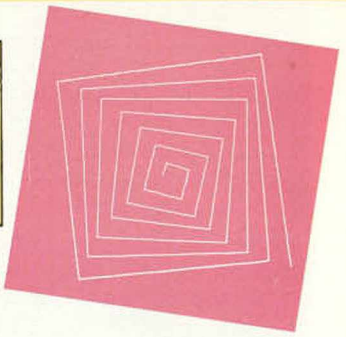
```
AP POLIESPI :COMPRIMENTO
  SE :COMPRIMENTO > 50 ENTAO PARE
  POLIESPI (:COMPRIMENTO + 15)
  FR :COMPRIMENTO
  DI 90
FIM
```

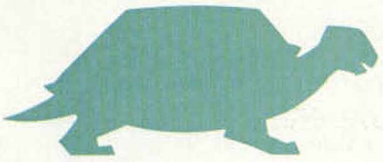
Ao rodar esse procedimento, você perceberá que a figura é desenhada "de trás para a frente": as linhas da espiral são traçadas da margem para o centro, e não o contrário. (Isso ficará mais claro se for usado um valor mais alto na sentença condicional — por exemplo, 250 em vez de 50.) O LOGO traça cada linha à medida que o controle é transferido pelas chamadas de procedimento. Nos exemplos anteriores, uma linha era traçada e então o controle passava a outro procedimento. Aqui, todos os procedimentos são chamados antes do início de qualquer desenho. Assim, usa-se antes o último valor de COMPRIMENTO introduzido.

Recursion (o termo mais usual para indicar a recorrência) é uma técnica que ocupa muito espaço na memória. Os procedimentos implementados com mais eficiência são aqueles em que a chamada recursiva está na última linha. Independente de quantas vezes sejam chamadas, elas não necessitam de memória adicional. Portanto, um procedimento recorrente só vale o trabalho de ser implementado quando é deste tipo.

Exercício 4

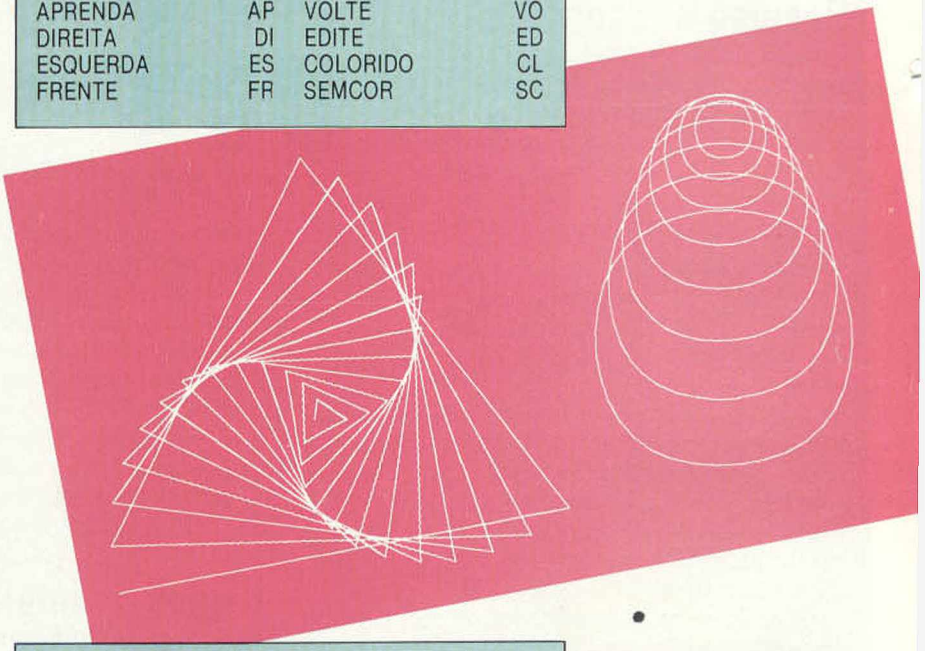
Escreva um procedimento para o desenho de uma torre formada de quadrados superpostos, de modo que a cada chamada o tamanho dos lados seja reduzido à metade.





Simplificação

APRENDA	AP	VOLTE	VO
DIREITA	DI	EDITE	ED
ESQUERDA	ES	COLORIDO	CL
FRENTE	FR	SEMCOR	SC



Respostas do exercício 3

A) Círculo

Este procedimento desenha um círculo, tendo o raio como um dado variável, e onde a posição inicial da tartaruga é um ponto na circunferência:

```
AP CIRCULO :RAIO
  REPITA 36 [FR(2* :PI* :RAIO/36) DI 10]
FIM
FACA "PI (3.14159)
```

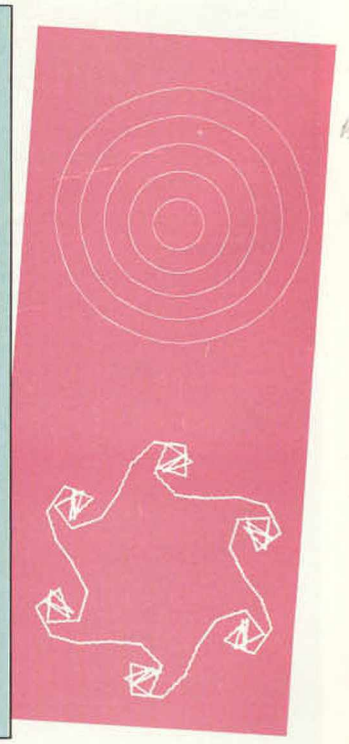
Se modificamos esse procedimento de modo que a posição inicial fique no centro do círculo, obtemos:

```
AP CENTRO.CIRCULO :RAIO
  SC ES 90 FR :RAIO DI 90 CL
  CIRCULO :RAIO
  SC ES 90 VO :RAIO DI 90 CL
FIM
```

B) Alvo

O ALVO emprega CENTRO.CIRCULO para desenhar cinco círculos concêntricos:

```
AP ALVO
  CENTRO.CIRCULO 10 CENTRO.CIRCULO 20
CENTRO.CIRCULO 30 CENTRO.CIRCULO 40
CENTRO.CIRCULO 50
FIM
```



CONCEITOS BÁSICOS

"Refogue o tempero." Ao dar essa instrução supondo que a leitora já saiba refogar, a receita lança mão de um algoritmo básico, ao invés de descrever todo um processo. Isso vale também para programas de computação.

Um algoritmo descreve qualquer processo em termos de outros processos anteriormente definidos (como rotinas já escritas no programa) ou tão básicos que dispensem definições (como comandos e funções preexistentes nas linguagens computacionais).

Algoritmos básicos formam outros, mais gerais, que, por sua vez, integram outros maiores, até que todo o programa seja escrito como um só algoritmo, formado por outros de nível inferior. Este é o conceito básico de programação estruturada ou modular.

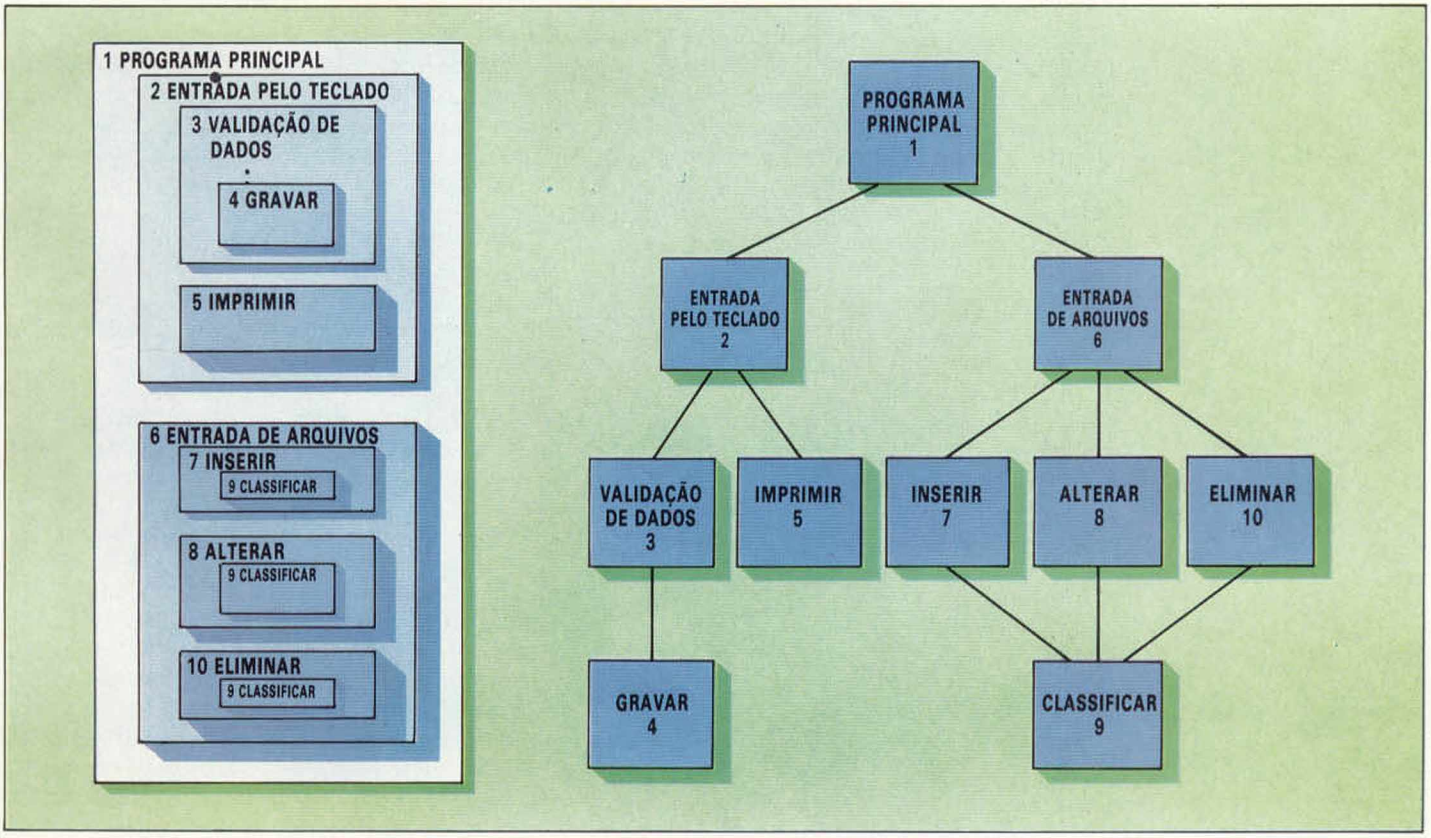
A matemática, a eletrônica, a estatística e outras ciências têm seus algoritmos, transportáveis para a linguagem computacional.

Todo algoritmo apresenta uma entrada e uma saída, isto é, trabalha com dados iniciais, ou parâmetros, para produzir resultados. Por exemplo: em LEFT$(P$,5), P$ e 5 são os parâmetros passados à função LEFT$. Esta, por sua vez, devolve um resultado em forma de string alfanumérico.

A base da elaboração de um algoritmo, portanto, reside na definição do conteúdo dos parâmetros de entrada e saída; seu tipo (inteiro, com decimais, alfanumérico etc.); e sua faixa de amplitude.

As instruções de um bom algoritmo caracterizam-se pela precisão (ausência de ambigüidades) e pela eficácia (não solicitando tarefas impossíveis, como o cálculo do maior número primo). O algoritmo precisa ser finito — apresentar testes eficientes para condições de saída — e não pode cair num loop sem fim. Suas boas qualidades incluem rapidez de execução e economia de memória. Convém, ainda, que seja simples e engenhoso ("elegante").

Importa bastante, também, a generalidade do algoritmo, sua capacidade de lidar com muitas situações diferentes daquela para a qual foi projetado. Desse modo, o programador monta uma biblioteca de rotinas reutilizáveis, como as que aceitam dados, as que mostram mensagens ao usuário, as que calculam funções etc.



Blocos e estruturas
O diagrama de blocos da esquerda mostra como os algoritmos de um programa se inserem uns nos outros. O fluxograma à direita realça as articulações entre eles e os níveis de hierarquia, no mesmo programa. Os algoritmos mais primitivos são os mais internos e mais baixos na hierarquia.

TERMINAL BANCÁRIO

No comando
Carlos Eduardo Correa da Fonseca, diretor-superintendente da Itautec, em 1985.

Voltada inicialmente para a montagem de sistemas bancários automatizados, a Itautec colocou-se entre as cinco maiores empresas nacionais de informática, produzindo a média de 1.500 equipamentos por mês.

Empresa do grupo financeiro Itaú, a Itaú Tecnologia S.A. (Itautec) foi fundada em novembro de 1979, com o projeto inicial de instalar os sistemas de agências *on line* (''em linha'', ou seja, controlados diretamente por um computador central) para o Banco Itaú, de modo a automatizar os serviços bancários. Seguindo o modelo americano do Citibank, que produz toda a rede de equipamentos necessários à montagem do sistema on line, um grupo de engenheiros do Itaú iniciou um trabalho de pesquisa tecnológica para o desenvolvimento do projeto.

Em junho de 1980, já se encontravam instalados cem terminais. Basicamente, o sistema BANKTEC consiste em terminais ligados por linhas telefônicas a um computador central, onde ficam armazenadas as informações relativas a cada cliente. O sistema conta com diversos equipamentos terminais: unidade de resposta audível, terminal caixa, terminal administrativo, terminal operado pelo cliente, terminal extrato operado pelo cliente e terminal transferência de fundos. Entre terminais e concentradores, a Itautec forneceu 7.000 equipamentos para a montagem do sistema on line do Banco Itaú. Essa marca foi conseguida graças à produção de duas fábricas — de circuitos impressos e de equipamentos de eletrônica digital. Fabricando a média de 1.500 unidades/mês, a indústria de equipamentos encontra-se instalada numa área de 8.000 m².

A fábrica de circuitos impressos ocupa área de 2.000 m² e atende tanto às necessidades da empresa como à crescente demanda do mercado de circuitos impressos.

Mediante convênios, a Itautec mantém intercâmbio de equipamentos, tecnologia e know-how com importantes universidades do país.

No final de 1982, a empresa lançou seu primeiro microcomputador, o I-7000 de 8 bits, destinado à aplicação profissional. Na mesma linha de microcomputadores, entraram no mercado versões simplificadas desse equipamento, o I-7000 Jr. e o I-7000 Jr. E — este só tem possibilidade de uma placa de expansão. O I-7000 Jr. é uma versão ainda mais simplificada para baratear os custos do produto. Não possui teclado numérico. Atende às necessidades de processamento de dados da pequena empresa e dos profissionais liberais.

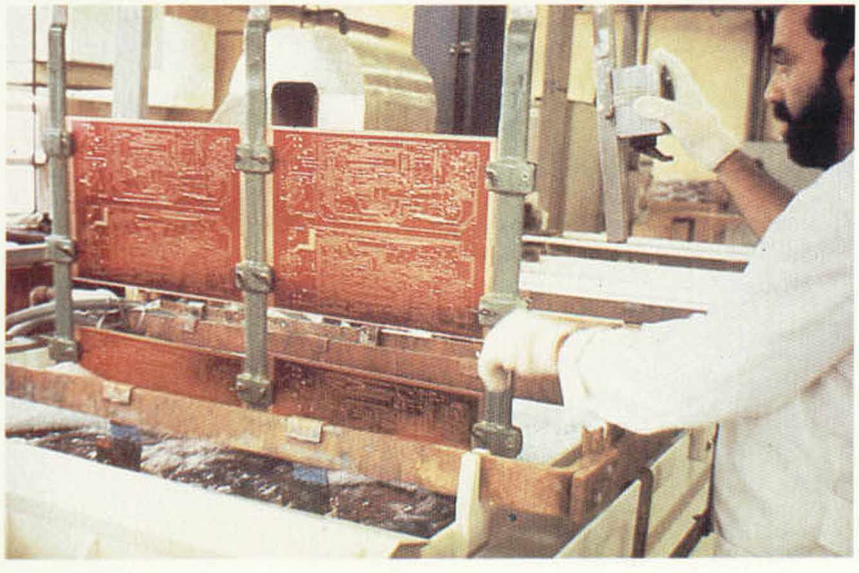
Em novembro de 1984, na Feira de Informática, foi lançado o I-7000 PCxt, de 16 bits, que contou com a experiência anterior, mas tomou por referência os PCs da IBM lançados nos Estados Unidos — o IBM PC e o IBM PCxt. O I-7000 PCxt da Itautec opera tanto com o sistema SIM/M quanto com o sistema SIM/DOS. Ou seja, é compatível com o I-7000 e com os IBM PC e PCxt.

Depois do lançamento do I-7000 PCxt, comercializado a partir de maio de 1985, a Itautec passa a desenvolver na mesma linha de tecnologia o IBM PCat americano, que será o próximo I-7000 a atender ao mercado nacional.

Na Feira de Informática de 1984 foi lançado também o IFAX 3021 — a primeira telecopiadora nacional. Funciona acoplada a telefone e, pelos circuitos de telecomunicação, transmite e reproduz de forma permanente textos, gráficos, ilustrações, desenhos e fotos. Essa transmissão se realiza pela rede nacional ou internacional de telecomunicações.

A Itautec é responsável também pelo lançamento de um dos primeiros computadores de médio porte brasileiros: o I-9000. Trata-se de um sistema completo de processamento de dados de 32 bits, compatível com IBM, o que permite o acesso a um grande conjunto de software disponível no mercado. A primeira unidade produzida foi vendida à Universidade de Campinas (Unicamp), em São Paulo, ainda em 1984.

Outro sistema desenvolvido pela Itautec é o videotexto. A empresa vem produzindo terminais, placas de expansão para microcomputador, concentradores telefônicos, equipamentos para gravação e atualização de dados e sistema de editoração. Com isso, a Itautec passa a ser a única a oferecer a solução completa para o videotexto — um sistema em que o usuário pode selecionar e acessar os mais variados tipos de informação utilizando a rede telefônica pública.



Indústria de componentes
Em seus 2.000 m² de área, a fábrica de circuitos impressos garante a produção para atender à crescente demanda do mercado de componentes.

Neste último caso, o programa passa ao próximo tipo de movimento e atualiza o contador de curvas. Se a linha encontrou um ponto iluminado passa-se ao fim do programa.

590-680: Desenha as quatro primeiras linhas, fazendo o contorno da tela.

690-780: Fim do jogo. Mostra quantas curvas foram feitas e toca a música de acordo com o número de curvas. Volta ao começo do loop principal para uma nova partida.

790-940: Apresenta os títulos e verifica se o usuário quer ver as instruções.

950-1080: Escreve as instruções na tela.

1090-1350: Inicializa rotina de som. Lê os dados das instruções DATA e faz o POKE (a inclusão) dos dados na área da memória reservada à variável A$. Note que, após rodar o programa, o valor atribuído a A$ será alterado. Note também que a versão cassete tem a linha 1220 alterada.

1280-1580: Rotinas de produção de sons. Você pode usar a rotina de inicialização e estas chamadas em outros programas de sua autoria.

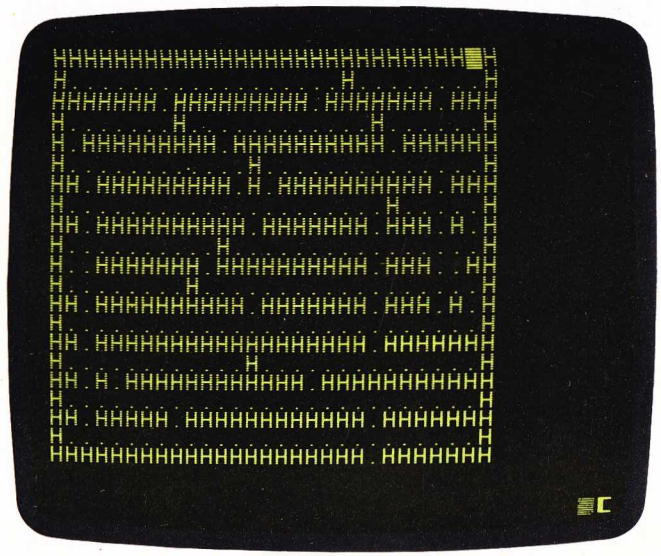
```
10   REM ********************************
20   REM *                              *
30   REM *   CRAM PARA LINHA TRS-80     *
40   REM *                              *
50   REM ********************************
60   REM
70   REM ********************************
80   REM *        INICIALIZACAO         *
90   REM ********************************
100  GOSUB 1100
110  DEFINT A - Z
120  GOSUB 810
130  C=0:I=0:J=2
140  REM ********************************
150  REM *       COMECO DO JOGO         *
160  REM ********************************
170  GOSUB 620
180  REM ********************************
190  REM *            LOOP              *
200  REM ********************************
210  REM
220  REM ********************************
230  REM *    MOVIMENTO PARA DIREITA    *
240  REM ********************************
250  I=I+1
260  IF POINT(I,J) THEN GOTO 720
270  SET(I,J)
280  IF INKEY$="" THEN 250
290  C=C+1
300  X=USR(8305)
310  REM ********************************
320  REM *     MOVIMENTO PARA BAIXO       *
330  REM ********************************
340  J=J+1
350  IF POINT(I,J) THEN GOTO 720
360  SET(I,J)
370  IF INKEY$="" THEN GOTO 340
380  C=C+1
390  X=USR(8319)
400  REM ********************************
410  REM *   MOVIMENTO PARA ESQUERDA      *
420  REM ********************************
430  I=I-1
440  IF POINT(I,J) THEN GOTO 720
450  SET(I,J)
460  IF INKEY$="" THEN GOTO 430
470  C=C+1
480  X=USR(8343)
490  REM ********************************
500  REM *     MOVIMENTO PARA CIMA        *
510  REM ********************************
520  J=J-1
530  IF POINT(I,J) THEN GOTO 720
540  SET(I,J)
550  IF INKEY$="" THEN GOTO 520
560  C=C+1
570  X=USR(8335)
580  GOTO 250
590  REM ********************************
600  REM *         DESENHA TELA           *
610  REM ********************************
620  CLS
630  FOR A=0 TO 127:SET(A,0):NEXT A
640  FOR A=0 TO 47:SET(127,A):NEXT A
650  FOR A=127 TO 0 STEP -1:SET(A,47):
     NEXT A
660  FOR A=47 TO 2 STEP -1:SET(0,A):
     NEXT A
670  A$=INKEY$
680  RETURN
690  REM ********************************
700  REM *          FIM DO JOGO           *
710  REM ********************************
720  GOSUB 1380 :IF C<10 THEN B$="! "
     ELSE B$="!"
730  PRINT @476, B$ ; C ; " !";
740  IF C=1 THEN C$="VOLTA" ELSE C$=
     "VOLTAS"
750  PRINT @540, "!"; C$ ; "!";
760  IF C>20 THEN GOSUB 1480
770  IF C>30 THEN GOSUB 1480 :GOSUB 1480
780  FOR A=1 TO 1500:NEXT:GOTO 130
790  REM ********************************
800  REM *             TITULOS            *
810  REM ********************************
820  CLS
830  FOR A=1 TO 15
840  PRINT TAB(6+A*2) "C R A M"
850  NEXT A
860  GOSUB 1380 :GOSUB 1380 :PRINT
     CHR$(23);
870  FOR B=1 TO 1500:NEXT B
880  CLS
890  PRINT TAB(8) "BENVINDO AO JOGO
     < CRAM > !"
900  PRINT TAB(8) "LIGUE O SEU MICRO A
     UM AMPLIFICADOR"
910  PRINT:PRINT "VOCE QUER INSTRUCOES?"
920  A$=INKEY$:IF A$ = "" THEN GOTO 920
930  IF A$="N" THEN GOTO 1070
940  IF A$<>"S" THEN GOTO 920
950  REM ********************************
960  REM *          INSTRUCOES            *
970  REM ********************************
980  PRINT:PRINT TAB(8) "O OBJETIVO DO
     JOGO E' VIRAR O MAIOR NUMERO"
990  PRINT TAB(6) "DE VEZES POSSIVEL NA
     TELA. CADA VEZ QUE VOCE"
1000 PRINT TAB(6) "QUISER MUDAR A
     DIRECAO DA LINHA QUE SE MOVE,"
1010 PRINT TAB(6) "USE QUALQUER TECLA
     EXCETO <SHIFT> OU <BREAK>."
1020 PRINT TAB(6) "O JOGO TERMINA QUANDO
     A LINHA BATER EM ALGUMA"
1030 PRINT TAB(6) "LINHA JA' DESENHADA.
     AO FINAL DO JOGO VOCE "
1040 PRINT TAB(6) "VERA' O NUMERO DE
     VEZES QUE CONSEGUIU DESVIAR."
1050 PRINT:PRINT TAB(14) "TECLE <ENTER>
     PARA COMECAR ..."
1060 A$=INKEY$:IF A$="" THEN GOTO 1060
1070 CLS
1080 RETURN
1090 REM ********************************
1100 REM *  INICIALIZA ROTINA DE SOM    *
1110 REM *      (A$ = 22 ESPACOS)       *
1120 REM ********************************
1130 AA$="                      "
1140 QA!=VARPTR (AA$)
1150 QB!=PEEK(QA!+1)+PEEK(QA!+2)*256
1160 REM ********************************
1170 REM *     PARA  VERSAO CASSETTE     *
1180 REM *     MUDAR LINHA 1220 PARA     *
1190 REM *     POKE 16526,PEEK(QA+1):    *
1200 REM *     POKE 16527,PEEK(QA+2)     *
1210 REM ********************************
1220 DEFUSR=QB!
1230 FOR A!=QB! TO QB!+21
1240 B!=A!:IF B!>32767 THEN B!=B!-65536
1250 READ Q:POKE B!,Q
1260 NEXT A
1270 RETURN
1280 REM ********************************
1290 REM *      LINGUAGEM DE MAQUINA      *
1300 REM *       PARA ROTINA DE SOM       *
1310 REM ********************************
1320 DATA 205,127,10,76,69,62,1
1330 DATA 211,255,16,254,69,62,16
1340 DATA 211,255,16,254,13,32,239,201
1350 REM ********************************
1360 REM *       SOM DO FIM DO JOGO       *
1370 REM ********************************
1380 FOR A=1 TO 3
1390 X = USR(8319)
1400 X = USR(8343)
1410 X = USR(8335)
1420 X = USR(8305)
1430 NEXT A
1440 RETURN
1450 REM ********************************
1460 REM *     SOM MAIS DE 20 VOLTAS      *
1470 REM ********************************
1480 X = USR(32383)
1490 X = USR(32383)
1500 X = USR(14479)
1510 X = USR(13719)
1520 X = USR(27287)
1530 X = USR(12966)
1540 X = USR(13719)
1550 X = USR(27287)
1560 X = USR(27287)
1570 X = USR(27287)
1580 RETURN
```

COME-COME

PARA A LINHA SINCLAIR

Você tem de percorrer o labirinto, comendo os pontos à medida em que avança. Em seu encalço, vem um monstro ameaçador. Você vence o jogo se conseguir percorrer todo o labirinto e apagar todos os pontos sem que o monstro o devore. Se o monstro for mais rápido, você perde a rodada. O programa vai construir, com as letras "H", as paredes do labirinto, e colocar nos corredores uma linha de pontos. Em seguida, alguns "H" e pontinhos são intercambiados para formar os caminhos internos do labirinto. Aparece então um cursor no canto superior esquerdo da tela. Este percorre o labirinto conforme você use estas teclas: 5 (que o movimenta para a esquerda), 6 (para baixo), 7 (para cima) e 8 (para a direita).

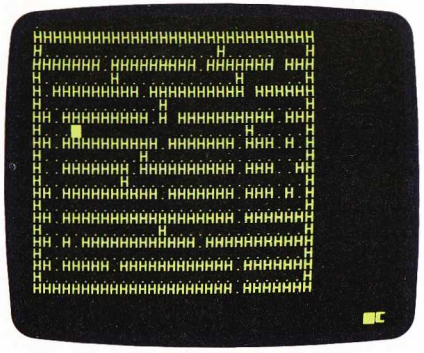
```
1    REM ********************************
2    REM ***** PARA A LINHA SINCLAIR *****
3    REM ********C O M E - C O M E *******
4    REM ********************************
5    REM *** O LABIRINTO ***
10   FAST
20   PRINT
25   REM *** CONSTRUCAO DAS PAREDES SEM
     SAIDA ***
30   FOR I= 1 TO 10
40   PRINT" HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
     H"
50   PRINT" H.............................
     H"
60   NEXT I
70   PRINT" HHHHHHHHHHHHHHHHHHHHHHHHHHHHHH
     H"
80   FOR N= 1 TO 30 STEP 14
90   FOR J = 3 TO 19 STEP 4
100  PRINT AT J,N;"."
110  NEXT J
120  NEXT N
125  REM *** DEFINICAO DOS CAMINHOS DENTRO
     DO LABIRINTO ***
130  FOR N= 5 TO 17 STEP 6
140  PRINT AT N,11;".";AT N,21;"."
150  NEXT N
160  PRINT AT 9,2;".";AT 9,30;"."
170  PRINT AT 13,2;".";AT 13,30;"."
180  PRINT AT 7,15;".";AT 7,17;"."
```



Começa a perseguição
Representado pelo cursor, você está prestes a iniciar sua corrida pelo labirinto. Cuidado com o monstro!

Um inimigo implacável
Para vencer o monstro, é preciso muita calma, além da rapidez. Neste caso, por exemplo, você ainda precisa comer todos os pontos do canto superior à direita.

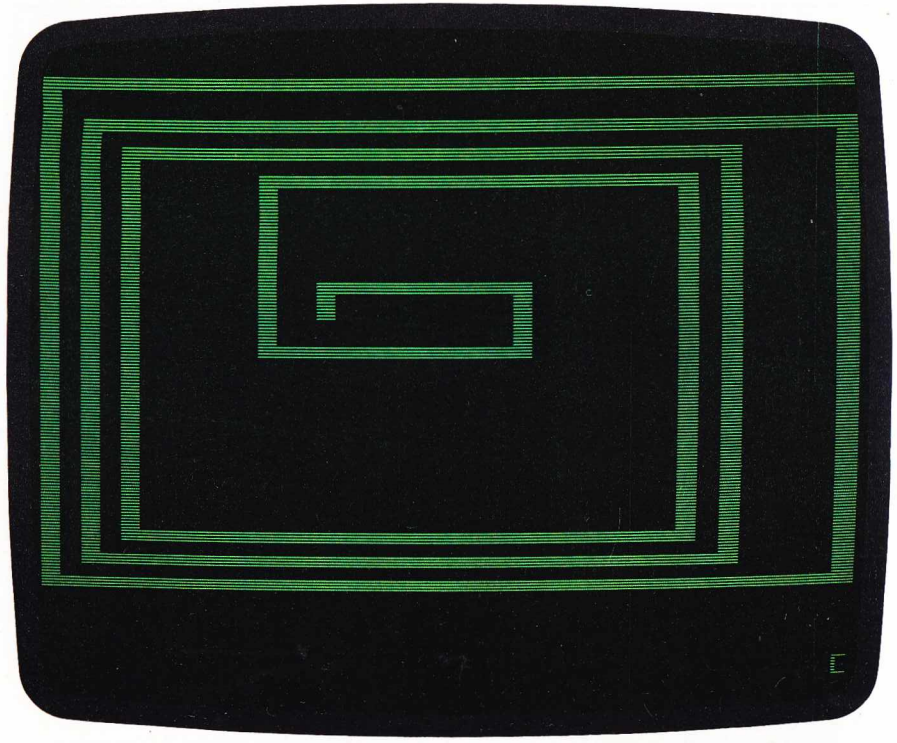
```
190 PRINT AT 15,15;".";AT 15,17;"."
200 PRINT AT 9,16;".";AT 13,16;"."
210 PRINT AT 11,3;".";AT 11,29;"."
220 PRINT AT 6,16;"H";AT 7,16;"H"
230 PRINT AT 11,16;"H";AT 15,16;"H"
240 PRINT AT 16,16;"H"
250 FOR N=1 TO 13 STEP 2
260 PRINT AT N,4;"."
270 PRINT AT N,28;"."
280 NEXT N
290 FOR N=8 TO 14 STEP 2
300 PRINT AT N,3;"H"
310 PRINT AT N,29;"H"
320 NEXT N
325 REM *** A ENTRADA NO LABIRINTO ***
330 PRINT AT 0,0;" "
340 LET C=PEEK 16398+PEEK 16399*255+115
350 LET P=C+495
360 LET S=27
370 SLOW
380 REM*** O JOGO ***
390 LET B=6.5
400 LET X=0
405 REM *** AGUARDA ENTRADA DO TECLADO
    ***
410 LET M$=INKEY$
420 IF M$<>"" THEN LET B=VAL M$
430 POKE C,S
440 POKE P,0
450 LET X=34*B-50*SGN (B-6.5)-221
460 IF PEEK (P+X)<>136 THEN LET P=P+X
465 *** MONTA O MONSTRO ***
470 POKE P,128
480 LET Y=33*SGN (INT ((P-C)/28))
490 IF PEEK (C+Y)<> 136 THEN LET C=C+Y
500 LET Y=SGN ((ABS ((P-C)/33-INT ((P-C)/
    33)) <.5)-.5)
510 IF PEEK (C+Y)<>136 THEN LET C=C+Y
520 LET S=PEEK C
525 REM *** O MONSTRO PERSEGUINDO ***
530 POKE C,151
535 REM *** VERIFICA SE O MONSTRO TE
    PEGOU ***
540 IF P=C THEN GOTO 560
550 GOTO 380
560 REM *** FINAL ***
570 CLS
```



```
580 PRINT AT 10,10;" TENTE DE NOVO, POR
    QUE DESTA VEZ!"
590 PRINT
600 FOR N= 1 TO 10
610 PRINT"        "
620 PRINT "NAO DEU"
630 NEXT N
640 PRINT "QUER TENTAR DE NOVO? S/N"
650 INPUT B$
660 IF B$="S" THEN GOTO 1
670 PRINT
680 PRINT
690 PRINT"TCHAU"
```


CRAM
PARA LINHA TRS-80

Neste jogo, o programa começa traçando uma linha que margeia a tela. Pouco antes de fechar a linha e completar o retângulo, você deve desviá-la (sempre para a direita), de modo a formar novo retângulo, interno ao primeiro. Procure fazer o maior número de retângulos e veja quantas vezes é capaz de desviar em tempo. O BASIC da linha TRS-80 Modelo III e seus compatíveis não incluem comandos de som; assim, há uma pequena rotina em linguagem de máquina responsável pela produção de efeitos sonoros. Caso seu microcomputador não possua alto-falante, você pode obter os efeitos ligando o cabo de saída do gravador a um amplificador.



Funções de trechos do programa

100-130: Inicializa a rotina de som, variáveis em geral, apresenta títulos e instruções, e atribui valores às variáveis principais — C (contador de curvas), I e J (coordenadas do ponto iluminado na tela).

180-580: Corpo principal do programa. Executa os quatro movimentos (direita, baixo, esquerda e cima) e verifica se a linha encontrou algum ponto iluminado (POINT I,J) e se alguma tecla foi usada (INKEY$).